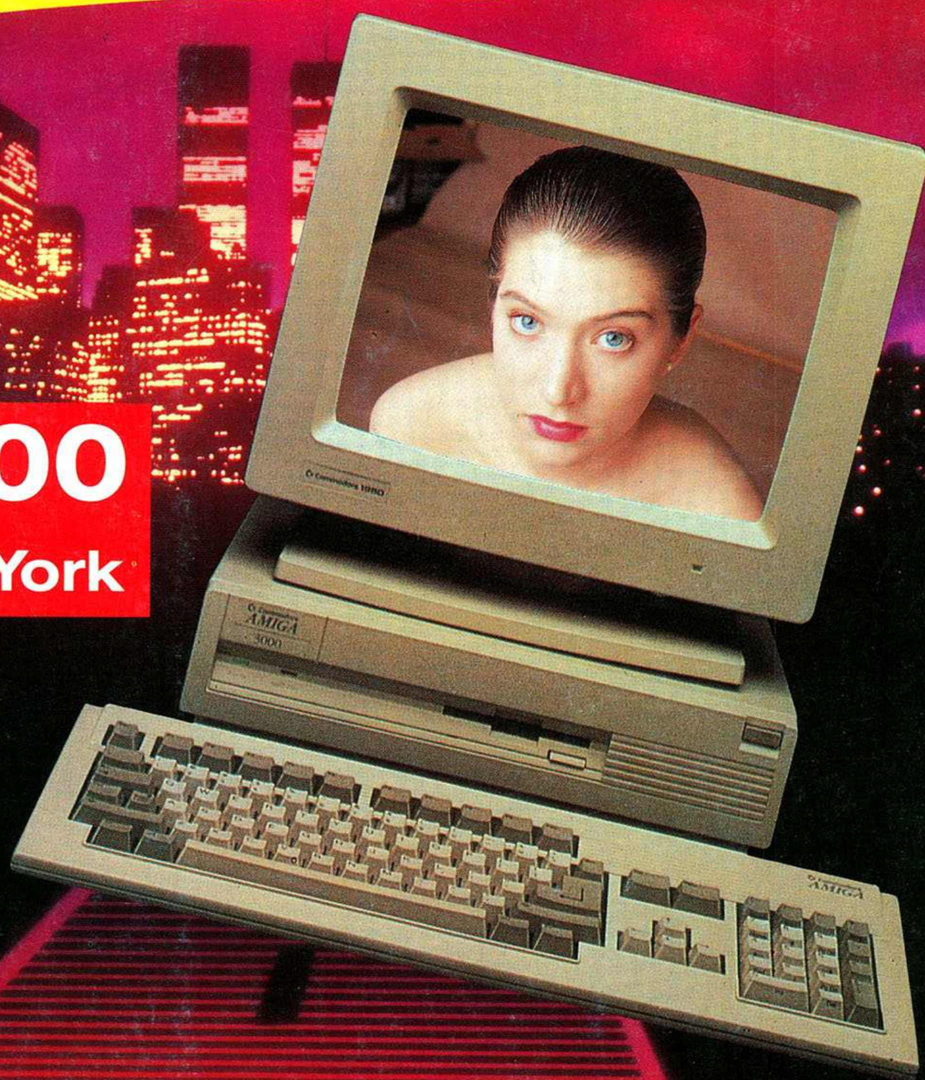
DAS MAGAZIN FÜR AKTIVE AMIGA-ANWENDER

DM 6,50
sfr 6,50
öS 52

7/90

AMIGA DOS

7/90
ISSN 0937-2717
DMV-Verlag



Amiga 3000
Aktuelles aus New York

Im Test
- Scanner
- Star LC24-15

Einsteiger-Kurse
- CLI
- Modula2

Textverarbeitung
- Grundlagen
- Programme

Tolle Tips und Tricks
- Externes Booten
- Rund um den Drucker

MIT GROSSEM SPIELE-TEIL

Hattrick!!!

FOOTBALL MANAGER
World Cup Edition



Kevin Toms hat mit seinen beiden Fußball-Manager-Simulationen einen Standard geschaffen. Football Manager II war eines der erfolgreichsten Programme 1989. Jetzt kommt die World Cup Edition!

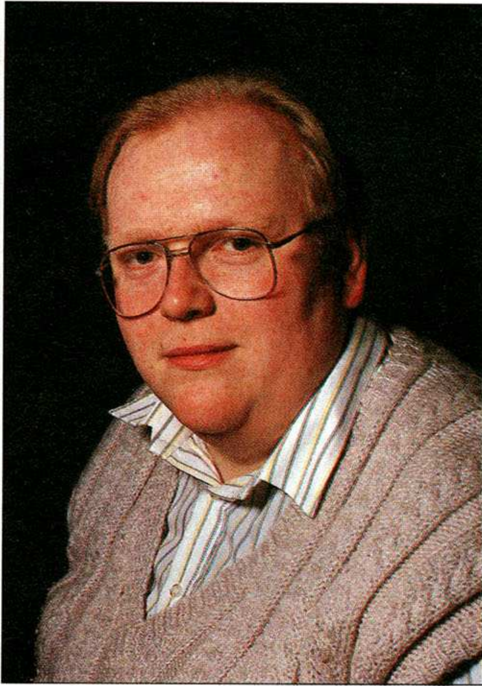
Das Meisterstück für AMIGA, ATARI ST, PC, C 64 und AMSTRAD CPC!



„Ich habe professionelle Manager interviewt und Fußball-Taktiken studiert. Das sind die Ergebnisse:
— mehr Realitätsnähe,
— mehr Spielbarkeit,
— mehr taktisches Management.
Ich habe mein Bestes getan, um das unterhaltsamste Spiel zur WM zu schaffen!“



Ariola Soft
Das Programm



Babylon – oder die Kunst zu programmieren

Es gibt Momente, in denen man sich wünscht, der Amiga würde alles, was man sagt, verstehen, ohne daß man selbst auch nur einmal die Tastatur berührt. "Geht doch", höre ich einige von Ihnen sagen, "wozu gibt es die Workbench!"

Schön und gut, aber, Hand aufs Herz, gefällt Ihnen alles, was Sie so an Fertigkost vorgesetzt bekommen? Oder existiert da nicht irgendwo der heimliche Wunsch, dieses oder jenes zu verbessern oder überhaupt etwas neu zu schaffen? Aber wie? Natürlich, programmieren heißt das Zauberwort. Und schon geht es los! Welches Spracherl hättens' denn gern? BASIC, PASCAL, MODULA, C, ASSEMBLER? Der angehende 'Programmdirektor' fällt in tiefe Verzweiflung; vor allem, wenn diverse 'Fachleute' noch zu diversen Sprachen raten. "BASIC programmiert doch keiner mehr!", "'C' ist was für Chaoten!", "Assembler? Bin ich ein Lemming?", Sprüche, die der Lernende sehr bald zu hören bekommt und die ihn doch keinen Schritt weiterbringen.

Wir von der AMIGA DOS wollen da anders sein. Wir wollen Ihnen helfen, sich auf dem Amiga weiterzubilden, den Computer zu einem Arbeitsmittel zu machen, das Sie unterstützt und nicht verwirrt.

Deshalb finden Sie bei uns Programme und Tips in allen gängigen Sprachen und sogar Kurse, wie den in dieser Ausgabe beginnenden Modula2-Kurs.

Aber nicht nur das, der Schwerpunkt dieser Ausgabe liegt auf dem Thema Textverarbeitungen, die zeigen, daß der Amiga auch in puncto Korrespondenz und Texterstellung einiges zu bieten hat.

Und, um wieder an den Anfang zurückzukehren, vielleicht ist es ja der neue Amiga 3000, den Sie ebenfalls in dieser Ausgabe kennenlernen, der endlich jedes Wort versteht, das ihm gesagt wird.

Oder ist es vielleicht doch besser, erst mal selbst 'sprechen' zu lernen?

In diesem Sinne

Jürgen Bornigel

INHALT

AMIGA NEWS

SOFTWARE

HARDWARE

TIPS & TRICKS

TITEL

KURS



Star Micronics setzt auf die Breite. Der Star LC 24-15 ist ein Drucker für den Privat- und Businessbereich. Den Testbericht lesen Sie auf

Seite 12



In dieser Ausgabe beginnt der große Modula-Kurs. Nicht nur Einsteiger, sondern auch Umsteiger werden gefordert. Lesen Sie unseren Bericht auf

Seite 69



Rainbow Islands – die Helden von Bubble Bobble sind wieder auf Tour. Diesmal erleben Sie zauberhafte Abenteuer auf den Regenbogeninseln. Wie es den Helden ergeht lesen Sie auf

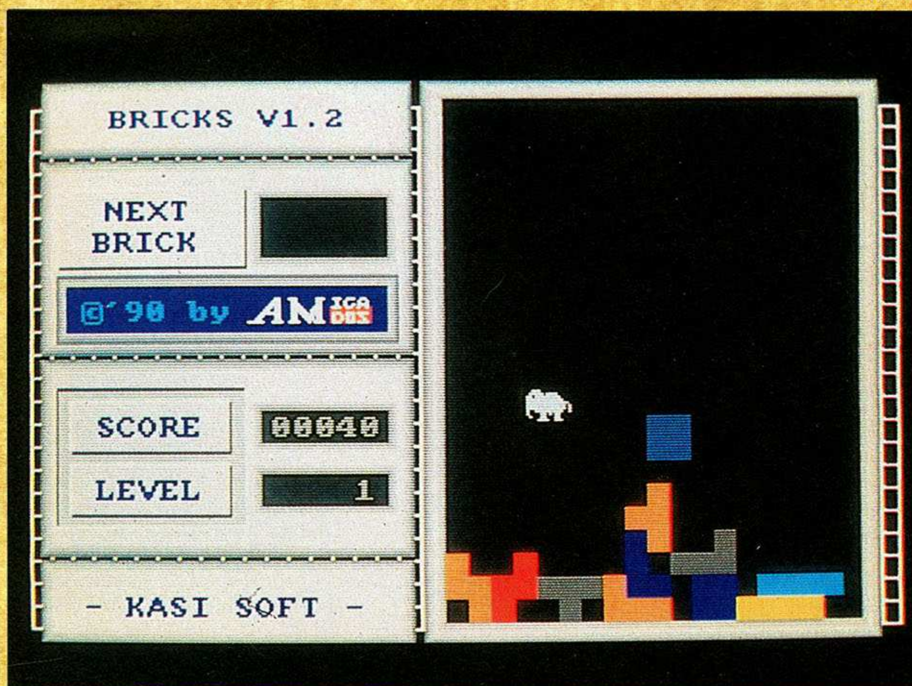
Seite 111

AMIGA DOS



Lang erhofft und heiß ersehnt: der Amiga 3000. Recht wundersame Gerüchte kursierten bisweilen durch die Gazetten. Commodore hat endlich den Mantel des Schweigens gelüftet. Über dieses Wunderwerk der Technik lesen sie Näheres auf

Seite 10



BRICKS – Nach dem Riesenerfolg von Tetris fand dieses Spiel überall seine Nachahmer. Wir stellen Ihnen eine sehr gelungene Tetris-Umsetzung vor. Lassen Sie sich von Logik und Kombination, gepaart mit Reaktion, gefangennehmen. Weiteres dazu finden Sie auf

Seite 76



Marshal M. Rosenthal

Eat-a-Bug!

Abenteuer Interaktion

An einem kalten und grauen Tag lenken wir unsere Schritte zum Marriott Marquis Hotel, das inmitten des Time Square Districts in Manhattan zu finden ist. Grund unseres Besuches ist die Interactive Conference...

Auf dieser Konferenz finden sich einmal im Jahr diejenigen ein, die sich in irgendeiner Weise mit interaktiven Computerprogrammen beschäftigen. Ob Hardwarespezialisten oder Softwareautoren, dieses Meeting dient regelmäßig dem Meinungsaustausch und der Vorstellung neuer Ideen zum Thema Interaktion. Aber auch bekannte Vertreter aus den Bereichen Video, Elektronik und Computerunterhaltung sind zusammengekommen, um ihre Vorstellungen von der Zukunft des interaktiven Mediums Computerspiel auszuarbeiten.

Eat-a-Bug – nicht nur ein Kinderspiel

Spezieller Grund unseres Besuches ist allerdings Dean Friedman mit seinem Video-Computer-Interface Eat-a-Bug. Die Vorstellung, einen Käfer zu verspeisen, hört sich zunächst einmal recht witzig an, bis man erfaßt hat, welche technologischen Möglichkeiten Bug bietet. Zuerst wurde dieses System im Rahmen einer Unterhaltungssendung Ende 1989 vorgestellt. Kinder hatten die bisher einzigartige Möglichkeit, in ein Videospiel hineingesetzt zu werden. Der Betrachter sah den Spieler in einen Grafik-Screen einkopiert, der eine Gartenszene zeigte. Dieser Screen füllte sich dann zusehends mit animierten Käfern, Bienen, Spinnen und Tausendfüßlern (hört sich ein bißchen wie ein bekanntes Atari-Spiel an...).

Das Spiel: Der auf 90 Sekunden eingestellte Timer startet den Countdown, und die niedlichen Tierchen beginnen sich zu bewegen. Jetzt wird es interessant. Jedesmal wenn der Spieler einen Käfer trifft, verschwindet dieser mit einem netten Soundeffekt. Erinnern wir uns: Der Spieler ist mit der Grafik auf dem Videoscreen "verbunden", und seine Handlungen haben Auswirkungen auf das Spiel, und das in Echtzeit! Aber da gibt es noch andere Effekte: Eine Biene fliegt heran und versucht, den Spieler zu stechen, eine Spinne versucht, sich auf dessen Kopf abzuseilen. Wenn eines der beiden Monster Erfolg hat, gibt es ein optisches und akustisches Signal, zum Beispiel ein platzender Ballon oder die Mundwerkzeuge der Spinne in Großaufnahme. Nach drei solchen "Unfällen" ist dann Game over.

Was wir hier etwas vereinfacht beschrieben haben, stellt eine sehr ausgefeilte Art und Weise der Interaktion zwischen Video und Computer dar. Und außerdem gibt es noch einige weitergehende Applikationen zu diesem System. Erklimmen wir also den siebenten Stock des Hotels, wo sich der Konferenzraum befindet.

High-Tech macht's möglich

Auf den ersten Blick glaubt man eher, in einem Mini-TV-Studio zu sein. In der einen Ecke finden wir Scheinwerfer und eine professionelle Videokamera, die an ein spezielles Schaltpult angeschlossen ist. Dies erlaubt die Auswahl und das Mischen mehrerer Videoquellen. Die gesamte Anlage erreicht allerdings Ausmaße, die sich kaum noch in einem normal dimensionierten Wohnzimmer unterbringen lassen. Ergänzt wird diese Anordnung durch einen unüberschaubaren Kabelwust, der eine Reihe von Monitoren und natürlich zwei Amigas mit Strom versorgt. Auf die anderen Peripheriegeräte wollen wir an dieser Stelle nicht näher eingehen. Dean Friedman steht mittlerweile vor einer Projektionswand und zieht wild gestikulierend die Aufmerksamkeit auf den Beginn der Vorführung.

Doch wenden wir uns erneut dem Schaltpult zu. Wir sehen eine Anzahl von verschiedenen Screens auf der Monitorreihe. Auf einem der Bildschirme sehen wir Eat-a-Bug-Grafiken mit laufender Animation, auf einem anderen sehen wir Dean. Diverse andere Screens zeigen Datazeilen an, die von dem BASIC-Programm benutzt werden. Ein weiterer Bildschirm stellt dann das Amiga-Programminterface mit seinen Gadgets und Schaltern dar. Der große Colormonitor hingegen bringt alle Bestandteile auf einen Schirm.

Stellt sich nur noch die Frage, ob Dean oder die zahlreich erschienenen Käfer den Kampf für sich entscheiden können. Lassen wir Dean Friedman zunächst das Spiel beenden. Während er nun etwas Atem schöpft, wollen wir einen Blick auf seinen Werdegang werfen.

Als gelernter Musiker konnte Dean in den siebziger Jahren eine Reihe von erfolgreichen Titeln in den Charts plazieren. Dann erwachte sein Inter-

esse, Musik mit Computern zu machen. MIDI war hier das Schlagwort. Von da aus war es nur ein kleiner Schritt, einen Amiga zu kaufen. Aber lassen wir ihn selbst zu Wort kommen:

"Mein erster Gedanke bei der Anschaffung eines Personalcomputers war, daß ich das Gerät für Musikeffekte nutzen kann. Ich hätte mich eigentlich für den Atari ST entschieden, aber ich konnte mich mit dem Keyboard einfach nicht anfreunden. So endete alles mehr oder weniger zufällig mit einem Amiga 1000, als dieser auf den Markt kam. Die Begeisterung stellte sich ein, sobald ich den Computer angeschlossen hatte.

Ich war völlig von den grafischen Fähigkeiten der Maschine überwältigt.

So begann meine Arbeit mit dem Amiga. Im Vordergrund stand die Idee, mit dem Amiga Musik zu machen. Nach und nach kamen Programmieren und Computerspiele dazu.

Mein drei Jahre altes Kind wollte natürlich auch den Amiga benutzen, und so begann ich, kleine Spiele zu programmieren. Ich stellte fest, daß es einen deutlichen Mangel an Spielen auf dem Amiga gab, die für Kinder geeignet waren.

Meine Grundidee bestand darin, das Ganze interaktiv zu gestalten, so daß der Spieler gezwungen ist, sich in Echtzeit mit dem Geschehen auseinanderzusetzen."

Beeinflußt hat ihn besonders Very Vivids Mandala-Programm. "Leider hat Very Vivid es nie geschafft, das System auf den Markt zu bringen. Aber einige Computerkünstler griffen die Idee auf. Das System war ursprünglich konzipiert, um sich mit Musik interaktiv auseinanderzusetzen. Das weckte natürlich meine Neugierde. Mandala war also bei der Umsetzung meiner Ideen richtungsweisend."

Nun muß man aber bedenken, daß alles auf 'Downstream'-Mixing ausgelegt ist. Das bedeutet, daß professionelles Fernsehstudio-Equipment zum Einsatz kommt. Damit kann der User in qualitativ hochwertiger High-Res-Grafik über den Spielscreen kopiert werden. Der Grund für die überragende Qualität liegt in der Verwendung von High-End-Equipment. Das Resultat ist mit dem vergleichbar, was der Fernsehzuschauer jeden Abend zu sehen bekommt. Das Verfahren, mit dem der Spieler nun in den Grafik-Screen eingesetzt wird, heißt Chroma-Keying. Bei diesem Verfahren wird der Spieler vom Hintergrund gelöst und dann mittels Superimposing in die Grafik kopiert. Dieses Verfahren verhindert jegliche Doppelbilder beziehungsweise Schatten. Der Spieler wird quasi auf den Grafik-Screen gestempelt. Andererseits kann man das System auch dazu benutzen, um zum Beispiel das Abbild des Spielers horizontal um 180 Grad zu drehen, so daß der Spieler ein Spiegelbild von sich selbst beim Spielen sehen kann.



Bild 1. Mit Hilfe dieser Hardwareanordnung wird Eat-a-Bug realisiert
Photography by Marshal M. Rosenthal



Bild 2. Mit Eat-a-Bug wird jedes Videospiel...
Photography by Marshal M. Rosenthal



Bild 3. ...zum Erlebnis, das mitunter etwas schmerzhaft ausfallen kann
Photography by Marshal M. Rosenthal

"Soweit zu diesem System – und jetzt zu meinem. Ich benutze zwei Amigas, die über einen MIDI-Anschluß miteinander verbunden sind. Der erste Amiga, ein A1000 mit 2,5 MByte RAM, steuert den zweiten Amiga, einen A500 mit 1 MByte RAM, indem er MIDI-Signale an den A500 sendet. Der 500er ist für das Abspielen der Musik und der Soundeffekte verantwortlich, die digital gesampelt und codiert sind. MIDI einsetzen heißt in diesem Fall, daß zwei Amigas problemlos zusammenarbeiten. Der A500 als Soundcomputer hat natürlich auch keine CPU-Verzögerung zur Folge, da er ja von allen anderen Aufgaben entlastet ist. Die Sounds wurden mit Hilfe von AMAS als Hardware und Audio Master 2 als Software erstellt.

In der Zwischenzeit bearbeitet der Hauptrechner das Script für Eat-a-Bug. Ein gutes Genlock führt die RGB-Signale vom Computer zum Videoschaltpult. Ebenso wird das Abbild des Spielers in das Genlock geführt.

Außerdem wird das Abbild des Spielers noch einmal in den Master-Amiga eingespeist, wo A-Squareds Live-Board, ein Echtzeit-Digitizer, eine weitere Bearbeitung vornimmt. Doch warum sieht er auf dem Screen nicht digitalisiert aus?

Die Antwort ist, daß ich Live auf etwas eigenwillige Weise benutze. Live zeigt dem Computer, wo im Raum sich der Spieler befindet, indem es den Spieler als Objekt 'rendert'. Diese Information wird dann für die Kollisionsabfrage benutzt. Im Prinzip heißt das, daß der Computer weiß, wo der Körper des Spielers



Bild 4. Die Devise heißt entweder, sich zu wehren...
Photography by Marshal M. Rosenthal



Bild 5. ...oder sich rechtzeitig aus der Gefahrenzone zu entfernen
Photography by Marshal M. Rosenthal

sich im Verhältnis zu den anderen Objekten auf dem Schirm befindet.

Alle Grafiken sind mit Hilfe von Dpaint III erstellt worden. Mandala als Interface bringt alle Komponenten zusammen auf einen Schirm. Aber auch dieses System ist nicht ganz fehlerfrei. Es lohnt sich aber trotzdem, auf diese Weise zu arbeiten, die Ergebnisse sind wirklich ausgezeichnet. Very Vivid hat Mandala hauptsächlich an Institute und Universitäten verkauft, eine Junior-Version für den Endverbraucherbereich soll im Spätsommer ` 1990 erhältlich sein."

Soweit zu den Vorgängen im Computer.

"Jetzt müssen wir nur noch verhindern, daß das digitalisierte Abbild sichtbar wird. Zur Darstellung gelangt also das Realbild, während der Computer intern mit dem digitalisierten Bild arbeitet. Das Script steuert nun die Grafik und Animationssequenzen und aktiviert diese. Die Ergebnisse können sich sehen lassen. Der Spieler erscheint in TV-Qualität und setzt sich mit Computergrafik auseinander, die auf den Spieler reagiert."

Dies eröffnet natürlich ungeheure Möglichkeiten für Geschäfte, Rundfunk und Präsentationen. Die Informationsvermittlung an Schulen und Büchereien könnte interaktiv gestaltet werden.

Auch Musikvideos finden hier ihren Platz, und es gibt weitreichende Möglichkeiten im Bereich Live-Action-Programmierung und Animation. Dazu kämen dann noch Fernsehshows. Die Wettervorhersage und sogar ein Kochkurs könnten auf diese Weise sehr lustig werden.

Aber auch der durchschnittliche Konsument mit dem ihm zugänglichen Equipment kann auf dieser Basis sehr gute Ergebnisse erzielen.

Dean erklärt weiter, daß Eat-a-Bug bewußt einfach gehalten ist, um einen Markt zu erreichen, der bisher in dieser Form noch nicht abgedeckt war.

"Ein anderes Spiel, das ich entwickelt habe, ist Toxic Waste. Da stehen Sie am Ufer eines Sees und versuchen alles herauszufischen, was an giftigen Dingen in den See geworfen wurde. Im Moment arbeite ich gerade an der Spielhallenversion eines High-End-Unterhaltungsgeräts. Stellen Sie sich nur den Spaß des Spielers vor, der sich selbst innerhalb eines Videospieles sieht. Wie er herumgeht, springt und Handlungen durchführt. Da kann man direkt ins Schwitzen kommen.

Aber es gibt noch so viele Möglichkeiten, die in Eat-a-Bug nicht enthalten waren. Zum Beispiel kann der Spieler Grafikobjekte mitnehmen, Dinge aufheben und sie benutzen, wie zum Beispiel Schwerter, Zaubertränke und dergleichen mehr. Es ist der physische Kontakt zwischen Realität und Videospielrealität, den ich zu erklären versuche. Stellen Sie sich einmal vor, wie realistisch und beeindruckend dann ein Karate-Spiel wäre!"

Vermutlich wäre der Spieler schnell von seinen bisherigen Videospielen gelangweilt. Aber versuchen Sie doch selbst mal eine Runde Eat-a-Bug.

Neue Abenteuer locken immer, und so ist es keine Frage, daß wir einige Versuche wagen. Drei Spiele später "schmerzt" das Hinterteil ein wenig von den Stichen der Bienen, interaktive Unterhaltung kann ganz schön anstrengend sein.

Dean verrät uns, daß bereits Verhandlungen mit Arcade-Maschinenherstellern im Gange sind. Möglicherweise wird es solch ein interaktives System schon im Spätsommer 1990 in New York geben.

Wir werden sehen...

(mm)

Neues von X-Pert

X-Pert bietet ab sofort eine 12-MHz-Version der Commodore-Brückenkarte A2088 für den A2000 an. Die Karte kann per Software jederzeit auf Tastendruck zwischen 4,77; 7; 11; 8,00 und 12 MHz umgeschaltet werden. Damit kann die Karte auch bei Hardware, die keine 12 MHz verträgt (zum Beispiel langsame Speichererweiterungen), eingesetzt werden.

Die Karte kostet inklusive Commodore-Lieferumfang 1198, – DM. Auch alte Karten können auf die 12-MHz-Version umgerüstet werden. Die Aufrüstung der PC-Karte auf 12 MHz kostet 398, – DM. Daneben können alle X-Pert-Karten mit 8 MHz für einen Update-Beitrag von 198, – DM auf 12 MHz aufgerüstet werden.

Info
X-Pert Computer Services GmbH
Weiherwiese 27
6270 Idstein/Ts.
Tel: 06126/3056 (8809)
Fax: 06126/5 49 22

Vectortrace in neuer Version

Kurz vor Redaktionsschluß erreichte uns von der Firma Gold Vision die Nachricht, daß das Programm Vectortrace in einer neuen Version 1.1 mit einigen neuen Features ab Mai im Handel erhältlich ist.

Neben den bisherigen Ausgabeformaten bietet die neue Version die Möglichkeiten, Grafiken im "Professional Draw-Clip-Format" und als IFF-Screenshot abzuspeichern. Daneben wurde die Arbeitsgeschwindigkeit um 20 Prozent gesteigert. Das Programm arbeitet mit neuen Requestern, wobei der Filerequester jetzt Tasten mit dem Namen der Laufwerke und eine "Parent"-Funktion besitzt.

Im Software-Review-Teil finden Sie eine ausführliche Beschreibung der Vectortrace-Version 1.0.

Info
Gold Vision
Kurfürstendamm 64-65
1000 Berlin 15
Tel: 030/88 33 505

Sicherheit durch Secu-Disk

Secu-Disk nennt sich ein neues Datensicherungssystem, das Datendieben das Handwerk legen soll. Es handelt sich hierbei um eine Diskette mit Schloß, die einfach in den Floppy-Schacht eingeschoben und abgeschlossen wird.

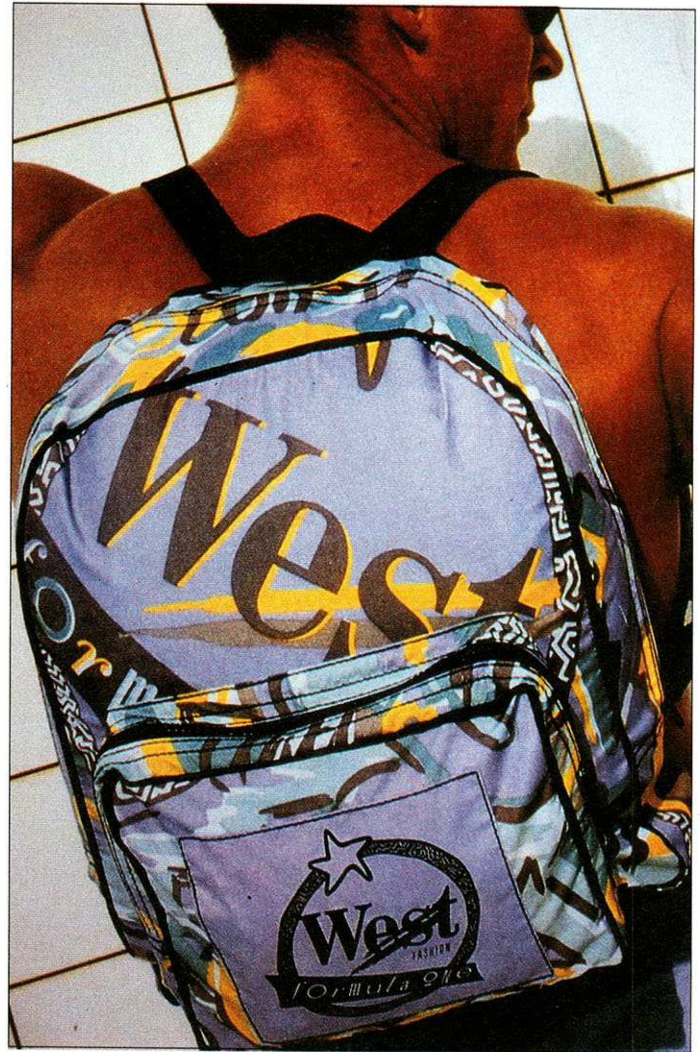
Damit soll auch die Infizierung eines Systems durch Computerviren ausgeschlossen werden. Secu-Disk kann selbst nach dem Startvorgang – sofern er nicht von Diskette erfolgt – im Laufwerk verbleiben.

Info:
FOX Computer & Peripherie Handelsgesellschaft mbH
Rheinstraße 13 - 23
6234 Hattersheim 3
Tel.: 06190/71071

Modisches für Ost und West

"Fun & Fantasy nicht nur auf dem Computer" ist das Motto für diesen Sommer. Auch Computerfreaks machen irgendwann einmal Urlaub. So richtig schön auffallen und sich von der breiten Masse abheben kann man mit der Sommerkollektion West Formula One Fashion. Knackige Urlaubsmode, nicht nur für Hacker und Computerfreaks, ist angesagt. Wer am Strand ohne Computer nicht auskommt, der findet mit einem speziellen West-Formula-One-Rucksack genau das Richtige. Hier kann man problemlos den Amiga-Laptop (vielleicht ist er ja bis dahin schon auf den Markt...) einpacken. Ein Extrafach für die Software ist auch vorhanden; was will der Computer-Enthusiast denn mehr? Man kann den Rucksack natürlich auch anders nutzen...

Info:
West: Fun & Fantasy Vertriebs GmbH
Parkstr. 51
2000 Hamburg 52



Formula-One-Rucksack – auf Schusters Rappen

Update von Digitiger

Die Firma Klaus Tute hat ein Update seiner Digitiger-Software herausgebracht. Damit ist es möglich, Bilder in 2 bis 64 Farben zu digitalisieren. Interessierte Digitiger-Besitzer können das Update zum Selbstkostenpreis beziehen. Neuere Besitzer erhalten das Update kostenlos. Leider hat sich in unserem Testbericht in der Ausgabe 6'90 ein qualitativ minderwertiges Foto eingeschlichen. Wir bitten dies zu entschuldigen.

Info
Fa. Klaus Tute
Mathildenstr. 12
3000 Hannover 91
Tel: 0511/62 98 25


Leichtes Digitalisieren mit Digitiger

Präzisionsjoystick

Ob Pacman oder Flugsimulator – jeder Spaß findet schlagartig sein Ende, wenn der Abgang durch einen unhandlichen und kontaktscheuen Joystick verursacht wird. Mit "Streichhölzern" läßt sich nun einmal kein Flugzeug steuern. Der Zubehörspezialist Lindy vertreibt neuerdings eine Serie von Präzisionsjoysticks, die diesbezüglich tatsächlich Freude am Spiel aufkommen lassen sollen. Ein solider Griff aus Stahl, ummantelt mit einem weichen Spezialgummi, das sich der Form der Hand anpaßt, soll für ein robustes Steuermedium sorgen. Über Drehschalter können die drei Feuerknöpfe beliebigen Funktionen zugeordnet und damit auf die Gewohnheiten des Anwenders abgestimmt werden. Auch der mechanische Widerstand des Griffs läßt sich individuell in neun Stufen regulieren. Sollte trotz der werkseitigen Voreinstellung ein Nachjustieren der Auswertungsmechanik nötig sein, kann dies mit einem mitgelieferten Schraubendreher problemlos über leicht zugängliche Stellschrauben durchgeführt werden. Derart ausgerüstet, eignet sich dieser Joystick auch für "ernsthafte" Anwendungen, wie beispielsweise im CAD-Bereich.

Info:
Lindy Elektronik GmbH
Postfach 102033
6800 Mannheim 1
Tel.: 0621/46005- 0



Multimedia Live!

Die Premiere des Amiga 3000

Gerüchte gab es schon lange um einen neuen Amiga. Auf der Cebit 1990 wäre er fast zu sehen gewesen. Aber am 24. April wurde nun in New York offiziell der Schleier gelüftet und der Amiga 3000 enthüllt. Hat sich das Warten gelohnt?

Multimedia ist das Stichwort, unter dem sich der neue Amiga verstanden wissen will. Sozusagen ein äußerst flexibler Alleskönner, ein Wanderer zwischen den Welten Busineßanwendung, Video, Präsentation und professionelle Soundanwendung.

Technik vom Feinsten

Es hat sich einiges verändert im neuen Amiga 3000, und geht man einmal von dem bisherigen Spitzenmodell, dem Amiga 2500/30 aus, nicht nur in technischer Hinsicht. Von Haus aus verrichtet ein Motorola 68030 seine Arbeit und wird von einem 68881- oder 68882-Coprozessor unterstützt. Der Datenbus ist auf 32-Bit-Basis konzipiert, was bedeutet, daß nun das gesamte System und nicht nur der Prozessor auf 32-Bit-Basis arbeitet. Mit einer Taktfrequenz von 16 oder 25 MHz, je nach Modell, ist für fulminante Geschwindigkeit gesorgt, die in vielen Fällen die Anschaffung eines Turbo-Boards nicht notwendig erscheinen läßt.

Das alles wäre wahrscheinlich nicht in diesem Maße möglich, wenn nicht insgesamt acht Coprozessoren den 68030 bei seiner Arbeit entlasten würden. Auch in Bezug auf Speicher hat der Amiga 3000 einiges zu bieten. So steht standardmäßig 1 MByte Chip-RAM zur Verfügung, das intern auf 2 MByte erweiterbar ist. Das Chip-RAM wird durch 1 MByte Fast-RAM ergänzt, das intern auf bis zu 16 MByte aufgestockt werden kann. Der Adreßbereich allerdings, und hier liegt ein weiterer Clou des Systems, erfaßt ein Gigabyte direkt adressierbaren Fast-RAMs!

Selbstverständlich wird das gesamte vorhandene RAM autokonfiguriert, die Devise heißt also: einstecken, einschalten, loslegen!

Ein 512-kByte-ROM auf 32-Bit-Basis stellt ebenfalls eine Neuheit dar.

Doch werfen wir auch einmal einen Blick auf die System-Slots. Ein 200-Pin-CPU-Slot mit 32 Bit unterstützt Cache-Memory-Karten oder einen weiteren schnelleren Prozessor, zum Beispiel den 68040. Auch am Amiga-System-Bus hat sich einiges getan, 16 beziehungsweise 32 Bit Erweiterungs-Slots Zorro III stehen zur Verfügung, die ebenfalls von Autoconfig unterstützt werden. Für den PC-Bus finden wir dann noch zwei PC-AT-Slots, die den Amiga 3000 mit der entsprechenden Brückenkarte auch für MS-DOS- und OS/2-Anwendungen verfügbar machen.

Ein 3,5-Zoll-Diskdrive und eine 19 ms schnelle 40-MByte-Harddisk stehen als Massenspeicher zur Verfügung, wobei die Festplatte bereits formatiert und mit der Systemsoftware versehen ist. Verbesserungen sind auch in den Grafikmöglichkeiten realisiert worden. So reicht nun die Palette von der Standardauflösung 320 x 200 bis 1280 x 400, was allerdings durch Anwendung der diversen Overscan-Modes noch gesteigert werden kann. Abhängig vom gewählten Grafikmodus können standardmäßig bis zu 4096 Farben dargestellt werden.

Der bereits im Amiga 2000 verfügbare Video-Slot ist auch im Amiga 3000 zu finden und bietet 12 Bit RGB-Analog und RGB-Digital. Direkt im Anschluß daran finden wir einen 100-Pin-Zorro-Slot, der speziell für die Integration von Zorro- und Video-Boards vorgesehen ist.

Besonders interessant ist, daß der Amiga an einer Vielzahl von Monitoren problemlos betrieben werden kann. So arbeitet er zum Beispiel mit

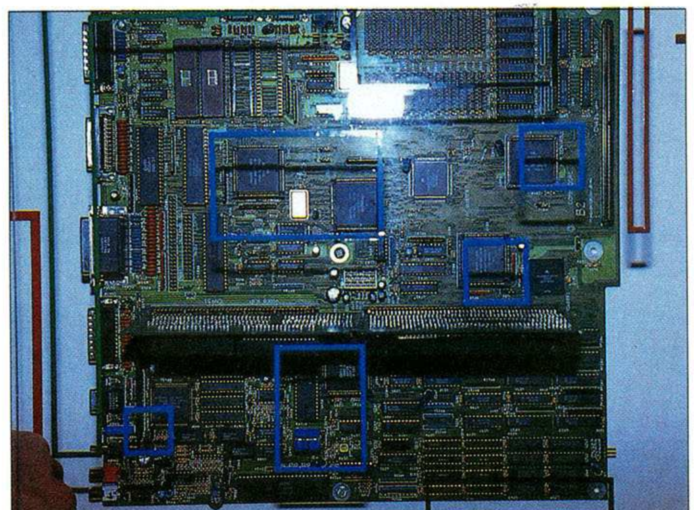


Bild 1. Das Motherboard des Amiga 3000...

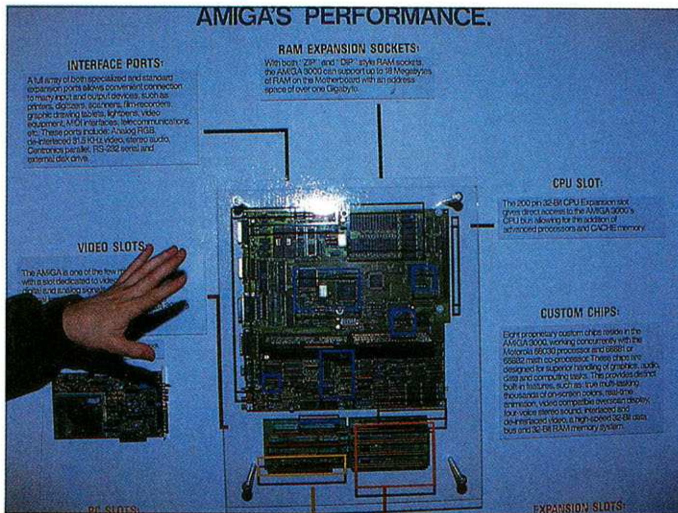


Bild 2. ...unterscheidet sich wesentlich von dem des A2000 wie man anhand der Coprozessoren erkennen kann

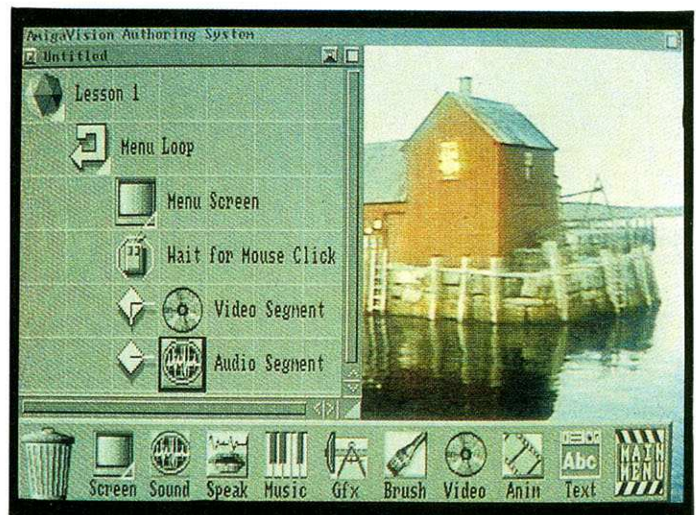


Bild 3. Das Betriebssystem Amiga-DOS V2.0 mit dem Amiga-Vision Authoring System vereinheitlicht die Bedienoberfläche

RGB-Analog-Monitoren oder auch mit VGA-Monitoren zusammen, auf denen allerdings nicht alle verfügbaren Modes unterstützt werden.

Die Soundmöglichkeiten des neuen Amiga sind ebenso beeindruckend. Vierstimmiger Stereosound bietet die Möglichkeit, auch komplexe Wellenformen zu reproduzieren. Der eingebaute Soundbuffer bietet nominal bis zu 800 kByte Speicherkapazität! Und wer gerne 'sampelt', braucht nicht mehr auf spezielle Hardware zurückzugreifen. Ein eingebauter 8-Bit-D/A-Wandler erledigt dies bereits in sehr guter Qualität.

Bleiben noch die Interfaces, die wohl kaum Wünsche offenlassen:

- Serielle Schnittstelle RS-232
- Parallelport Centronics
- 15,75 kHz Video (RGB-Analog oder RGB-Digital)
- 31,5 kHz non interlaced oder scandoubled RGB-Video
- links und rechts Audio-Stereo
- interne und externe Floppy-Disk-Anschlüsse
- Interne und externe SCSI-Ports
- Keyboard-, Maus-, Joystick- und Lightpen-Anschluß

Drei verschiedene Versionen des Amiga 3000 stehen ab Start der Serie zur Verfügung:

- A3000 mit 16 MHz, 68030 Prozessor, mathematischem Coprozessor 68881 und 19-ms-40-MByte-Festplatte
- A3000 mit 25 MHz, mathematischem Coprozessor 68882 und 19-ms-40-MByte-Festplatte
- A3000 mit 25 MHz, mathematischem Coprozessor 68882 und 19-ms-100-MByte-Festplatte

Die preisliche Gestaltung macht den Amiga 3000 nicht nur für Multis interessant, sondern rückt den jüngsten Amiga-Sproß deutlich auch in die Reichweite des ambitionierten Endverbrauchers. Leider stand die genaue Preislage für Deutschland bei Redaktionsschluß noch nicht definitiv fest. Der Preis dürfte (hoffentlich) nicht allzusehr von den Preisen abweichen, die in den USA für den Amiga 3000 verlangt werden.

Der Preisrahmen bewegt sich dort zwischen 3299,– US$ für die 16-MHz-Version, 3999,– US$ für die 25-MHz-Version und 4499,– US$ für die 25-MHz-Version mit 100-MByte-Festplatte. Ein äußerst interessanter Preis, wenn man bedenkt, was alles bereits serienmäßig enthalten ist.

Amiga-DOS V2.0 – Das neue Betriebssystem für den A3000

Ausgestattet mit dem neuen Betriebssystem Amiga-DOS 2.0 steht nun eine Benutzeroberfläche zur Verfügung, die die Möglichkeiten der Hardware bestens ausnutzt. Endlich ist es gelungen, neben einem wesentlich verbesserten optischen Eindruck einen Standard für alle System-Utilities, Icons, Requester und Gadgets zu schaffen. Dies führt natürlich zu einem vereinfachten und verbessertem Handling, sowohl für den User als auch für den Programmierer. Besonders interessant für Programmierer dürfte sein, daß nun auch standardmäßig Arexx in das neue Betriebssystem integriert wurde. Damit hat der Benutzer von Amiga-DOS V2.0 die Möglichkeit, sämtliche externen Prozesse, die ebenfalls Arexx unterstützen, von seinem Amiga aus fernzusteuern. Zusammen mit dem neuen Amiga-Vision-System steht nun ein Multitaskingsystem für weitreichende Multimedia-Präsentationsmöglichkeiten zur Verfügung. Nehmen wir einmal an, Sie planen die Vorstellung eines neuen Produktes. Mit Hilfe von D-Paint erstellen Sie eine Animation Ihres Firmenlogos. Als Ergebnis erhalten Sie nach Abschluß der grafischen Arbeit ein Icon, das die gesamte Animation repräsentiert. Ebenso wird mit dem Sound verfahren. Eine zum Produkt passende Musik wird mit Hilfe eines Editors erstellt oder der Einfachheit halber 'gesampelt' und in ein Stand-Alone-File umgewandelt. Auch hier resultiert die Arbeit wieder in einem Icon, das, je nach Bedarf, hinter oder neben das Animations-Icon gestellt wird. Auf Wunsch können Sie nun die Produktbeschreibung von Ihrem Amiga sprechen lassen oder weitere Grafiken und Sound erstellen. Jeder fertiggestellte Bereich wird nun nur noch durch ein Icon dargestellt, und um die gesamte Animation mit Sound abzuspielen, reicht ein Mausklick.

In der Tat ist dieses neue User-Interface so logisch und einfach aufgebaut, daß man ohne langes Studium der neuen Features sofort mit der Arbeit beginnen kann. (mm)



Bild 4. Und so präsentiert sich der Amiga 3000 im neuen Gewand

Big-Business-Druck

Büroatmosphäre mit dem Star LC24-15

Neue Drucker zu testen, ist fast eine undankbare Aufgabe für einen Redakteur. Die Unterschiede bei den neuen Geräten sind nicht mehr so offensichtlich wie noch vor ein paar Jahren. Der 24-Nadel-Standard hat sich langsam, aber sicher durchgesetzt, jeder Anwender stellt deutliche Ansprüche an seinen zukünftigen Drucker. In die Kategorie der 'Arbeitstiere' fällt auch der Star LC24-15. Wir haben ihn uns einmal genauer angesehen.



Bild 1. Der Star LC24-15 ist ein DIN-A3-Drucker für gehobene Ansprüche

Der Star LC24-15 ist eigentlich die DIN-A3-Ausgabe des schon bekannten Star LC 24-10. Das Bedienfeld des Druckers und der Netzschalter sind an der Vorderseite angebracht und gut zu erreichen. An Schnittstellen besitzt der LC24-15 einen Paralleleingang (Centronics) und einen Modulschacht für Speicher- oder Schriftartmodule. Der Parallelport sitzt in einem Interface-Steckmodul, das bei Bedarf gegen ein serielles Interface ausgetauscht werden kann. Diese Lösung kann man als optimal bezeichnen, gestattet sie doch den Einsatz des Druckers an unterschiedlichen Datenübertragungen, ohne das lästige Einbauen anderer Schnittstellen.

Der Drucker besitzt einen automatischen Einzelblatteinzug und eine Papierparkfunktion. Diese erlaubt das Bearbeiten von Einzelblättern, während sich noch Endlospapier im Drucker befindet. Ein schnelles Wechseln zwischen beiden Papiereinzugsarten ist somit ebenfalls möglich.

Die Mechanik des Druckers ist robust, der Druckkopf wird durch einen Rundstab und eine Zahnschiene geführt. Das Farbband wird wie üblich auf die Kopfhalterung gedrückt. Die Geschwindigkeit des Druckers ist recht beachtlich, 167 Zeichen pro Sekunde im Draft-Pica-Modus, 200 Zeichen pro Sekunde im Elite-Modus, 55 Zeichen pro Sekunde im LQ-Pica-Modus und 67 Zeichen pro Sekunde im LQ-Elite-Modus sind laut Herstellerangaben möglich, im Test zeigte sich der LC24-15 tatsächlich rasant. Die Dip-Schalter des Druckers sind ebenfalls leicht zu erreichen, eine kleine Klappe unter der Kopfführung muß abgehoben werden, dann liegen beide Schalterreihen offen. Die Dip-Schalter selbst müssen vorsichtig mit einem Kugelschreiber oder einer Plastikpinzette eingestellt werden, das übliche Problem also bei derartigen elektronischen Teilen.

Die Tastatur enthält alle Tasten, um Zeichensätze, Papiervorschub und Parkfunktion zu initialisieren. Interessant ist die Möglichkeit des Mikrovorschubs; hier kann durch eine Tastenkombination der Vorschub in kleinsten Schritten vorwärts und rückwärts gesteuert werden. Ein genaues Positionieren per Hand gehört damit erst einmal der Vergangenheit an.

Der Einzelblatteinzug des Star LC24-15 arbeitet präzise und zuverlässig. Einzelblätter werden bis zum Anschlag eingeschoben und danach per Tastendruck richtig positioniert. Auch nach mehreren Blattzuführungen befand sich der Druckkopf noch an der richtigen Position.

Endlospapier kann zusammen mit Einzelblättern zugeführt werden, dabei ist die Papierparkfunktion sehr hilfreich. Durch Tastendruck kann das Endlospapier zurückgesetzt werden, während die Einzelblätter bedruckt werden können. Die Papierzuführung endete allen Befürchtungen zum Trotz nicht mit reißwolfartigen Szenen; im Gegenteil: Drucker und Papierzuführung arbeiten "Hand in Hand".

Der **Star LC24-15** ist ein neuer DIN-A3-Drucker für den professionellen Einsatz. Seine Vorzüge sind **Schnelligkeit**, sein **automatischer Einzelblatteinzug**, sowie die **Papierpark-Funktion.** Geschrieben werden kann normal, **fett**, unterstrichen, *kursiv*, hochgestellt oder tiefgestellt, oder in einer **Kombination *aus allem!!!***

A

Der **Star** LC24-15 ist ein neuer DIN-A3-Drucker für den professionellen Einsatz. Seine Vorzüge sind **Schnelligkeit**, sein **automatischer Einzelblatteinzug**, sowie die **Papierpark-Funktion.** Geschrieben werden kann normal, **fett**, unterstrichen, *kursiv*, hochgestellt oder tiefgestellt, oder in einer **Kombination *aus allem!!!***

A

Abb. 1. Der Testausdruck mittels einer Textverarbeitung. Unten sehen Sie den gleichen Ausdruck in der Schriftart 'Script'



Abb. 2. Grafikdruck liegt dem LC24-15 nur in Verbindung mit einem guten Druckprogramm, hier unter Zuhilfenahme der Hardcopy-Funktion von Turbo Print II

Der Druckkopf ist ausreichend gekühlt, selbst bei Dauerbetrieb war keine Beeinträchtigung festzustellen. Die Geräuschentwicklung hält sich in Grenzen, beim Vergleich mit 24-Nadel-Kollegen schnitt der LC24-15 eindeutig besser ab.

Ein Gerät für Vieldrucker und Schönschriftliebhaber

Im weiteren Testverlauf wurden dem Kandidaten unter Preferences mehrere Druckertreiber vorgeschlagen, die er teilweise nicht akzeptierte. Die Einstellungen 'generic' und 'custom' brachten zwar ansehnliche Ergebnisse, was den Textausdruck betraf, Grafikausdruck ist mit diesen Treibern jedoch nicht zu schaffen. Erst der Treiber 'EPSON Q' brachte erste Ergebnisse. Der Grafikausdruck ist jedoch auch hier problematisch, da die Nadelsteuerung die Grafik aus Mustern zusammensetzt. Erst die Zusammenarbeit mit speziellen Druckerprogrammen wie Turbo Print II oder Perfect-Printer lieferten das, was man von einem Grafikausdruck erwartet. Beim Ausdruck von Geschäftsgrafiken ist es daher überlegenswert, ob man nicht auch noch ein bißchen in spezielle Grafiktreiber investieren soll, wie sie zum Beispiel Turbo Print II oder Turbo Print Professional liefern.

Das Zusammenspiel mit Textverarbeitungen klappte dagegen auf Anhieb. Alle bekannten Schriften wurden anstandslos übernommen und in guter Qualität umgesetzt. Mit eigenen Schriftarten ist der LC24-15 auch gut bestückt; neben 'Courier', 'Prestige' und 'Orator' gibt es noch 'Script', eine etwas ungewöhnliche, aber trotzdem saubere Schriftart. Sie stehen dem Anwender in Draft und Letter Quality zur Verfügung. Der LC24-15 ist ein typisches Arbeitstier, schnell und leise, gut ausgestattet, funktionell und ergonomisch. Er eignet sich vor allem als Drucker für Viel- und Schönschreiber, im Dauereinsatz beim Grafikdruck muß man Abstriche machen. Ansonsten sind keine Mängel zu melden, der Star macht seinem Namen Ehre.

(jb)

AMIGA DOS
Blitzlicht

Name: LC24-15
Hersteller: Star Micronics
Typ: 24-Nadel-Drucker für DIN-A3-Format
Preis: zirka 1200 DM
Anbieter: Fachhandel

Positiv:
- schnell
- mehrere Schriftarten
- 11 kByte Pufferspeicher
- aufrüstbar bis 32 kByte
- andere Schriftarten auf Modulen erhältlich
- austauschbare Schnittstellenmodule
- automatischer Papiereinzug

Negativ:
- durch die Preferences-Treiber schlechter Grafikausdruck



Andreas Polk

Bügeleisen für Bilder

'Vectortrace' vektorisiert Bilder

Ein unliebsamer Effekt beim Ausdruck von Bitmap-IFF-Bildern ist der Treppcheneffekt. Obwohl der Drucker in der Lage ist, Bilder in einer sehr hohen Auflösung zu drucken, werden im Bild nur die Pixel aneinandergereiht, die somit einzeln sichtbar sind. Ein Weg, dieses Problem zu umgehen, ist die Vektorisierung von Grafiken.

Was bedeutet Vektorisierung? Es gibt zwei Arten von Grafiken. Die bekannteste ist die Bitmapgrafik, wie sie beispielsweise von Deluxe Paint oder ähnlichen Malprogrammen erzeugt wird. Hier wird jeder einzelne Punkt der Grafik im Speicher festgehalten. Somit hat man nie mehr Informationen über das Bild zur Verfügung, als sich im Speicher Daten befinden. Anders ist dies bei Vektorgrafiken. Hier wird nicht das ganze Bild im Speicher gehalten, sondern Objekte werden durch Beziehungen zwischen Punkten definiert. Beispielsweise würde eine Linie bei einer Vektorgrafik nur zwei Angaben brauchen, die Start- und die Endkoordinate. Ein Malprogramm dagegen würde die Linie als Grafik im Speicher behalten und nicht die Koordinaten. Beide Methoden haben Vor- und Nachteile. Der Vorteil einer Bitmapgrafik liegt in ihrer bequemen Verarbeitung. Der Bildaufbau ist recht schnell. Bei einer Vektorgrafik muß dagegen die gesamte Grafik immer neu aufgebaut werden, wenn ein Objekt eingefügt würde. Das kostet natürlich Zeit. Dafür kann man eine Vektorgrafik sehr flexibel verarbeiten. Ein Objekt kann durch Veränderung seiner Koordinaten verändert werden. Unter anderem entfällt auch beim Druck der Treppcheneffekt, da das Druckprogramm die Auflösung der Grafik an die höchste Auflösung des Druckers anpaßt.



Bild 1. Vectortrace – ein Vektorisierungsprogramm

Vectortrace macht Grafiken runder

Genau hier setzt Vectortrace an. Es konvertiert eine Bitmapgrafik (also ein IFF-Bild) in eine Vektorgrafik. Diese Grafik kann dann in verschiedenen Formaten zwecks Weiterverarbeitung abgespeichert werden. Insgesamt gibt es vier Formate: Aegisdraw, Postscript, Videoscape und Vectortrace. Wird eine Grafik im Aegisdraw-Format abgespeichert, so kann Sie in den gängigen DTP-Programmen, wie beispielsweise Pagestream, weiterverarbeitet werden. Eine Postscript-Datei können Sie mit einem Laserdrucker oder mit Pixelscript in einer sehr guten Qualität ausdrucken. Ein Leckerbissen ist dann noch das Abspeichern im Videoscape-Format. So lassen sich aus Bildern ganz einfach zweidimensionale Objekte erstellen. Diese müssen aber noch per Hand weiterverarbeitet werden. Bei der Weiterverarbeitung aller Formate gab es keinerlei Probleme. Die Dateien wurden sowohl von Videoscape als auch von Pixelscript ohne Umschweife akzeptiert.

Das Konvertieren von Grafikdateien

Natürlich muß erst einmal ein IFF-Bild in eine Vektorgrafik konvertiert werden, was natürlich nicht ganz ohne Aufwand möglich ist. Vectortrace erledigt diese Arbeit aber in einer wirklich akzeptablen Zeit. Wählt man den Menüpunkt 'Erkennen' an, so fragt Vectortrace nach, ob das Objekt hinterher nur aus Linien oder auch aus Flächen bestehen soll. Haben Sie sich hier entschieden, so beginnt der Konvertierungsvorgang. Obwohl Vectortrace hierzu mehrere Durchgänge braucht, verbringt man relativ wenig Zeit wartend vor dem Bildschirm.



Abb. 1. Ein Beispiel, gedruckt als IFF-Bild...



Abb. 2. ...und als Postscript-Datei mit Pixelscript

Für komplexe Bilder werden höchstens ein paar Minuten gebraucht, einfache Bilder sind in weniger als einer Minute konvertiert. Hier kann eine Turbokarte die Geschwindigkeit bestimmt noch erheblich steigern. Neben diesen Grundfunktionen bietet Vectortrace noch die Möglichkeit, das Bild vor und nach der Konvertierung komfortabel zu verändern. So kann ein IFF-Bild beispielsweise invertiert werden, Streupunkte können unterbunden werden und Linien lassen sich verdicken. Hat man das Bild einmal vektorisiert, so können Punkte gelöscht, verschoben und hinzugefügt werden. Auch im nachhinein läßt sich noch die Farbe eines Objektes ändern. Dargestellt wird das gesamte Objekt wahlweise auf einem eigenen Screen im Interlace-Modus oder mittlerer Auflösung oder aber in einem Window auf dem Workbench-Screen. Dies spart am meisten Speicherplatz. Möchte man sich ein Objekt genauer anschauen, so kann man sich die Grafik frei wählbar vergrößern lassen. So können auch im Detail Verbesserungen vorgenommen werden.

Vectortrace ist ein gelungenes Programm. Es bietet auf einfache und komfortable Weise eine Lösung für das altbekannte Treppchenproblem. Daß das Programm Vectortrace alleine nichts nützt, kann dem Programm nicht als Nachteil angerechnet werden. Als einziger Minuspunkt wäre zu nennen, daß es schön wäre, dreidimensionale Objekte für Videoscape zu erzeugen. Hierzu müßte nur eine einfache 'Extrude'-Funktion eingebaut werden. Auch wäre es wünschenswert, daß die Objekte im Turbosilver- und Sculpt-4D-Format abspeicherbar sind. Fazit: Empfehlenswert für alle, die bei der Weiterverarbeitung von Grafiken nicht auf eine gute Qualität verzichten wollen.

(jb)

AMIGA DOS
Blitzlicht

Name: Vectortrace
Anbieter: Gold Vision, Kurfürstendamm 64-65, 1000 Berlin 15
Tel.: 030/8833505
Preis: 148,– DM

Positiv:
- schnelle Konvertierung
- vielfältige Manipulationen möglich
- mehrere Ausgabeformate
- gute Ergebnisse
- deutsches Handbuch

Negativ:
- keine Ausgabe im Turbosilver- und Sculpt-4D-Format



Andreas Polk

Digiworks 3D

Dimensionen haben keine Grenzen

Animationsprogramme üben allein durch ihre optischen Ausführungen einen ungeheuren Reiz auf den Betrachter aus. Wie die Umwandlung der verschiedenen Dimensionen machbar wird, zeigt dieser Test.

Digiworks 3D nennt sich ein Programm, das von den Animationsspezialisten Heinrichson, Schneider & Young vertrieben wird. Dieses Programm hat die Aufgabe, aus einem zweidimensionalen Bild ein dreidimensionales Objekt für Animationsprogramme zu erstellen. Auf diese Art und Weise lassen sich sehr einfach Objekte erstellen. Sie müssen nur mit einem Malprogramm gezeichnet werden, und Digiworks 3D konvertiert anschließend diese Bilder in Objekte für Sculpt-4D oder Turbosilver 3.0. Soweit der Anspruch des Programms. Kann es aber den hohen Ansprüchen gerecht werden?

Lieferumfang

Ausgeliefert wird Digiworks 3D in einer großen Pappschachtel. In besagter Schachtel befinden sich ein zirka 40 Seiten umfassendes englisches Handbuch und eine Diskette mit Digiworks 3D und einigen Demo-Bildern. Das Handbuch ist didaktisch gut aufgebaut und erfüllt seine Aufgabe. Allerdings sollte Heinrichson, Schneider & Young als Direktdistributor auch einmal an die nicht englischsprechenden Anwender denken und eine Übersetzung des Handbuchs ausliefern. So viel Arbeit dürfte das ja nicht machen. Das Handbuch führt in vier Tutorials an Hand von praktischen Beispielen in die Arbeit mit Digiworks 3D ein. Abgerundet wird es durch einen Referenzteil, in dem noch einmal alle Menüpunkte geordnet erläutert werden.

Hat man das Handbuch durchgearbeitet, so erkennt man auch schon, für welchen Anwendungsbereich Digiworks 3D konzipiert wurde: die Erstellung von dreidimensionalen Texten. Access Technologies brachte vor einiger Zeit die Fancy-3D-Fonts auf den Markt. Dies sind Zeichensätze von guter Qualität für Animationsprogramme, wie Sie auch in dieser Ausgabe an anderer Stelle lesen können. Diese Fonts wurden mit Digiworks 3D erstellt. Die Nachfrage nach diesem hausinternen Programm war so groß, daß man sich entschied, es zu überarbeiten und zu vermarkten. Doch nicht nur Schriften, sondern auch andere Objekte lassen sich mit Digiworks 3D erstellen.

Zuerst wird ein Bild im IFF-Format und möglichst nur mit zwei Farben benötigt. Dieses Bild wird in Digiworks 3D eingeladen und das Bild sofort abgescannt. Digiworks 3D sucht signifikante Punkte heraus und verbindet diese mit Linien. Damit haben Sie bereits einen ungefähren Überblick über das spätere Objekt. Hier ergeben sich allerdings auch schon direkt zwei Nachteile. Zum einen lädt Digiworks 3D Bilder, die nicht genau über das Format 640/320*283 Pixel verfügen, falsch ein, und zweitens ist es durch das direkte Starten des Scan-Vorgangs nicht möglich, sich ein Bild erst noch einmal anzuschauen, bevor man es konvertiert. Hat man nämlich das falsche Bild geladen, so muß man erst den ganzen Scan-Vorgang abwarten, was bei komplexen Bildern schon dauern kann. Ansonsten geht der Scan-Vorgang recht zügig vonstatten.

Konvertierung

Der Anwender kann sich nun entscheiden, ob er das ganze Bild in ein Objekt konvertieren möchte oder nur einen Teil. Ist dies der Fall, so kann nach Anwahl des Menüpunktes "Area" dieser Bereich festgelegt werden. Nur noch der entsprechende Bildausschnitt wird angezeigt und bearbeitet. Auch hier ergibt sich ein Nachteil: Möchte man wieder ein ganzes Bild bearbeiten, so muß man erst einen genügend großen Bildbereich ausschneiden und neu festlegen. Wünschenswert wäre hier ein Menüpunkt wie "Reset Area".

Editieren

Meistens entspricht das Ergebnis der Konvertierung nicht genau dem Ausgangsbild. Deshalb bietet Digiworks 3D dem Anwender einige nützliche Editiermöglichkeiten. So kann man Punkte verschieben, zum bereits bestehenden Objekt hinzufügen oder löschen. Auch ganze Linien können gelöscht werden. Eine leistungsfähige Funktion, die man meist nur von guten Editoren kennt, ist die Draw-Funktion. Mit ihr ist es möglich, mit ein paar Mausklicken ganze Linienzüge zu erzeugen. Klicken Sie einfach den Ausgangspunkt an und die weiteren Eckpunkte. Digiworks 3D verbindet diese direkt mit einer Linie.

Dieser Mini-Editor erfüllt seine Dienste gut, und die Objekte sind wirklich leicht nachzubearbeiten. Einziger Nachteil ist die Tatsache, daß keine Zoomfunktion vorhanden ist. So kann der Editiervorgang bei recht eng aneinanderliegenden Punkten auch sehr schwer, wenn nicht gar unmöglich werden. Digiworks 3D bietet jedoch nicht nur Funktionen zur Bearbeitung von Objekten nach dem Scannen an. Auch vor dem Einladen eines Bildes und dem anschließenden Scannen sind Optionen einstellbar, die das Ergebnis des Scan-Vorgangs beeinflussen. So kann man festlegen, wie dicht die Punkte an dem ursprünglichen Bild liegen müssen oder ab wann zwei in der Nähe liegende Punkte zu einem zusammengefaßt werden. All diese Einstellungen werden dann beim nächsten Scan-Vorgang berücksichtigt. Möchten Sie

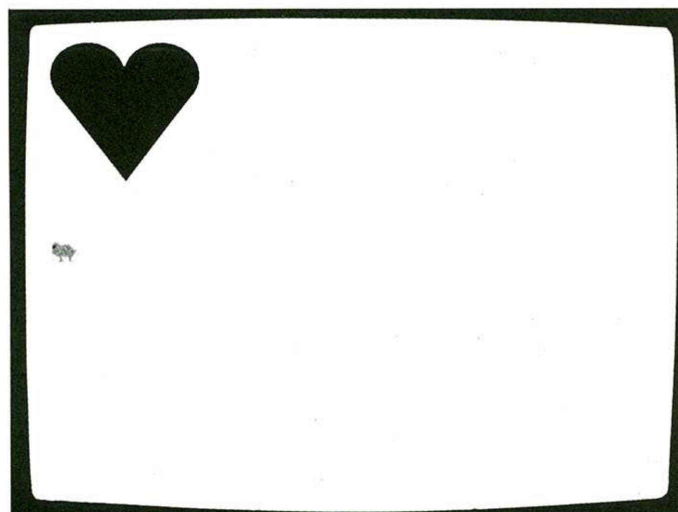


Bild 1. Das IFF-Bild...

AMIGA DOS Info

Die Programme Digiworks 3D und Vectortrace dienen dem Konvertieren von Grafiken in Vektorgrafiken bzw. Objekte für Animationsprogramme. Was unterscheidet diese Programme grundsätzlich, nach welchen Kriterien kann man eine Auswahl treffen, welches Programm für den Anwender das richtige ist?
Zuerst die grundsätzlichen Unterscheidungsmerkmale:
Digiworks 3D ist zum Erstellen von 3D-Objekten für Animationsprogramme zuständig. Zumindest Sculpt-4D wird dabei hervorragend unterstützt. Vectortrace wird von Goldvision vertrieben, die ihren Schwerpunkt auf DTP setzen. Hierfür eignet sich Vectortrace besser. Mit diesem Programm lassen sich Grafiken in den gängigen Formaten ausgeben, die von DTP-Programmen benutzt werden.
Die Ausgabe im Videoscape-Format kann als Bonbon angesehen werden. Allerdings ist für Videoscape-Besitzer Vectortrace hier wieder besser geeignet als Digiworks 3D, es sei denn, Sie besitzen auch noch Interchange. Also für Animationen (außer Videoscape): Digiworks 3D
Für DTP und Videoscape: Vectortrace.

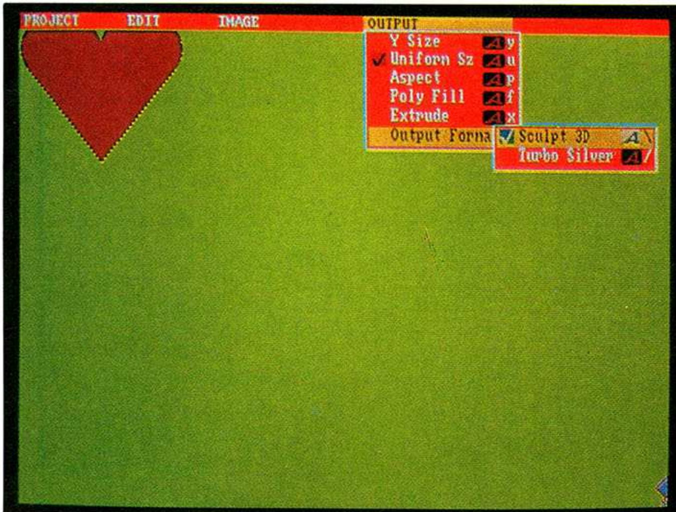


Bild 2. ...wird mit Digiworks 3D bearbeitet...



Bild 3. ...und anschließend berechnet

ein bereits eingeladenes Bild noch einmal berechnen lassen, so wählen Sie dazu einfach den Menüpunkt "Recalc" aus, das Bild braucht dann nicht mehr neu eingeladen zu werden.

Wurde nun die Umrandung des Objekts zufriedenstellend festgelegt, so muß Digiworks 3D nur noch die einzelnen Polygone erzeugen. Dazu wird die entsprechende Funktion aktiviert, und Digiworks 3D beginnt seine Arbeit. Sind allerdings einige Flächen nicht geschlossen, so protestiert Digiworks 3D lautstark. Sie haben die Möglichkeit, den Vorgang abzubrechen oder so weit ausführen zu lassen, bis Digiworks 3D zu dem ansprechenden Polygon kommt. So läßt sich immer leicht erkennen, wo der Fehler liegt. Allerdings wäre es schon eine erhebliche Erleichterung, wenn die entsprechenden Eckpunkte in einer anderen Farbe dargestellt würden, denn sonst kann die Suche recht lange dauern.

Beste Ergebnisse liefert Sculpt-4D

Zuletzt muß nur noch das Ausgabeformat und die Dimension des Objekts festgelegt werden. Weiterhin lassen sich auch die Oberflächeneigenschaften und die Farbe der Seiten- und Grundflächen einstellen. Hier kommt jedoch lediglich der Sculpt-4D-Anwender so richtig zum Zuge. Alle Einstellungen werden richtig interpretiert, und der Anwender muß nur noch die Umgebung festlegen. Bei Turbosilver 3.0 ist dies anders. Entweder werden alle Flächen in der Farbe Weiß oder gar nicht erzeugt. Hier hilft dem Anwender nur der Einsatz eines Konvertierungsprogramms, wie beispielsweise das Programm Interchange, oder die Einstellungen müssen per Hand vorgenommen werden. Leider erzeugt Digiworks 3D auch nur Objekte in diesen beiden Formaten, andere Programme wie beispielsweise Videoscape 3D oder Reflections werden nicht unterstützt.

Fertige Sculpt-4D- oder Turbosilver-3.0-Objekte können nicht mehr in Digiworks 3D eingeladen werden. Möchten Sie ein Objekt zu einem späteren Zeitpunkt noch einmal weiter bearbeiten, so müssen Sie es zusätzlich im Digiworks-3D-eigenen Format abspeichern. Dies ist allerdings kein Nachteil. Neben der Farbe und den Oberflächeneigenschaften kann noch festgelegt werden, wie tief das Objekt sein soll, wie weit es also in den Raum ausgebreitet werden soll. Ein ganz besonderes Bonbon bietet Digiworks 3D noch denjenigen, die Fonts erstellen wollen. Aktiviert man die Einstellung "Uniform Size", so werden alle Objekte gleich groß abgespeichert, was natürlich bei Schriften unumgänglich ist. Stellen Sie sich einmal vor, das 'a' wäre doppelt so groß wie das 'e'!

Zum Schluß beleuchtet

Faßt man die Funktionen und deren Aufgaben zusammen, so kommt man zu einem recht eindeutigen Ergebnis. Mit Digiworks 3D lassen sich sehr einfach zweifarbige Bilder in dreidimensionale Objekte verwandeln. Der Hauptanwendungsbereich liegt in der Erstellung von Fonts oder von Firmenlogos. Allerdings kommt man auch hier um die nötige Zusatzsoftware nicht herum. Dies kann nicht unbedingt dem Programm angerechnet werden, da Digiworks 3D kein autarkes Programm ist, sondern in eine Programmkette eingereiht werden muß. So braucht man beispielsweise ein Malprogramm, um die zu konvertierenden Bilder erst einmal zu erstellen. Auch ein Nachbearbeitungsprogramm à la Butcher kann oft hilfreich sein. Ganz entfalten kann sich Digiworks 3D allerdings erst in Zusammenarbeit mir einem Scanner oder einem Digitizer. Erst so ist die Schnittstelle zur Außenwelt hervorragend gegeben.

Professionelles Arbeiten ist möglich

Beispielsweise könnten Sie Ihr Firmenlogo oder eine interessante Schrift einscannen oder digitalisieren, mit Butcher oder D-Paint III nachbearbeiten und mit Digiworks 3D in ein dreidimensionales Objekt verwandeln. Für diesen Anwendungsbereich ist das Programm hervorragend geeignet. Lediglich Kleinigkeiten wären zu ergänzen wie beispielsweise die fehlende Lupenfunktion oder das manuelle Starten des Scanvorgangs. Wer von Digiworks 3D allerdings erwartet, daß das prächtig gemalte 32farbige Haus in Null komma nichts in ein ebenso schönes Objekt verwandelt wird, der hat sich getäuscht; das ist nicht möglich (aber auch nicht mit anderer Software!). Diejenigen, die Sculpt-4D nicht anwenden, könnten eventuell auch etwas enttäuscht sein. Sie müssen sich erst ein Konvertierungsprogramm zulegen, um die Objekte in verschiedene Formate zu konvertieren. Dies bedeutet noch einmal eine Anschaffung von über 100,- DM. Ansonsten ist Digiworks 3D zu empfehlen und gehört in die Programmsammlung eines jeden Animators.

(vb)

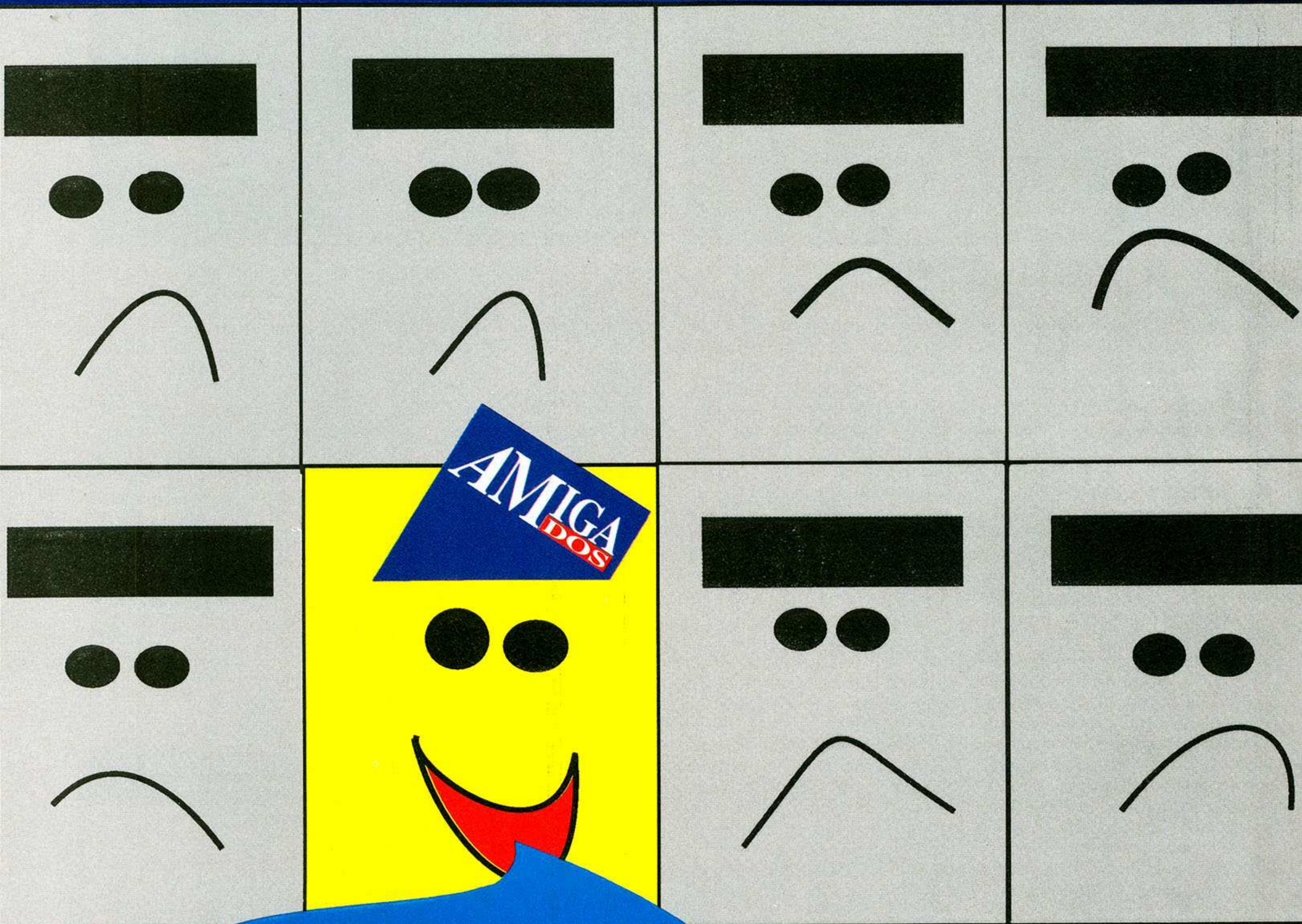
AMIGA DOS Blitzlicht

Name: Digiworks 3D
Preis: 248,- DM
Hersteller: Heinrichson, Schneider & Young
Classen-Kappelmannstr. 24
5000 Köln 41
Tel.: 0211/402365

Positiv:
- schneller Konvertierungsvorgang
- gute Editiermöglichkeiten
- vielfältige Voreinstellungen möglich
- arbeitet hervorragend mit Sculpt-4D zusammen
- gutes Handbuch
- kein Kopierschutz

Negativ:
- keine Zoomfunktion
- Scanvorgang startet automatisch
- teilweise kleine Programmfehler
- mangelhafte Unterstützung anderer Animationsprogramme
- Handbuch nur in Englisch

ABONNEMENT



INFORMATIONEN AUS ERSTER HAND

Ein Abonnement ist praktisch und bequem.

Widerrufsrecht
Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

AMIGA DOS
kostet im Abonnement:

Im Inland bzw. West-Berlin:
6 Ausg. = 35,– DM
12 Ausg. = 70,– DM

Im europäischen Ausland:
6 Ausg. = 50,– DM
12 Ausg. = 100,– DM

Im außereuropäischen Ausland:
6 Ausg. = 60,– DM
12 Ausg. = 120,– DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag



Andreas Polk

Datenbank für Profis?

Entwicklerpaket Superbase Professional

Die Version 3 des Entwicklerpakets Superbase Professional bietet gegenüber der letzten Version einige Verbesserungen und stellt eine gute Basis für die anspruchsvolle Datenverwaltung, -bearbeitung und -auswertung dar.

Spielen Sie mit dem Gedanken, sich eine Dateiverwaltung zuzulegen, so haben Sie die Möglichkeit, sich für eine einfache Dateiverwaltung, wie beispielsweise "Datamat" oder "Superbase" zu entscheiden. Damit lassen sich Daten archivieren und nach bestimmten Kriterien auswerten. Mehr war aber bisher kaum möglich. Etwas anders ist es bei den Entwicklerpaketen, die eine eigene Sprache zum Programmieren von Datenbanken bieten; sie sind natürlich entsprechend teurer. Dazu zählen Datamat Professional und Superbase Professional, im folgenden nur noch SB Professional genannt, das hier genauer unter die Lupe genommen werden soll. SB Professional verfügt über weitaus mehr Features, die über eine einfache Datensammlung hinausgehen.

So besteht beispielsweise die Möglichkeit, mehrere Dateien miteinander über gemeinsame Felder zu verknüpfen (relationale Datenverwaltung), Daten mit Hilfe zahlreicher Funktionen auszuwerten, mit Hilfe eines Filters nur bestimmte Daten herauszusuchen, mit der mitgelieferten Programmiersprache selbst Programme für häufige Auswertungs- oder Bearbeitungsroutinen zu schreiben und vieles mehr. Mit einem Preis von knapp 600 DM zielt das Programm auf den professionellen Markt und ist damit rund 200 DM teurer als das vergleichbare Datamat Professional. Lohnt sich eigentlich solch eine Mehrausgabe?

Lieferumfang

Das Paket wird in den von Markt & Technik gewohnten Pappkartons ausgeliefert. Das Handbuch ist ein ebenfalls gewohnter Ordner und umfaßt über 600 Seiten. Dies erscheint auf den ersten Blick sehr viel, ist aber bei näherer Betrachtung doch etwas knapp, da oft Beispiele fehlen und der Einsteiger allein gelassen wird. Weiterhin befinden sich im Ordner drei Disketten. Die erste enthält das Programm selbst, die zweite den Formulareditor und die dritte einige Beispielanwendungen (Dateien, Formulare, Programme in der mitgelieferten Programmiersprache und Bilder).

Unangenehm fällt dem Käufer beim Öffnen des Kartons ein kleiner schwarzer Stecker auf, der sich als Dongle herausstellt und somit als Kopierschutz fungiert. Ohne diesen Stecker, der im zweiten Joystickport seinen Platz findet, ist die Benutzung von SB Professional nicht möglich, was dem Anwender spätestens nach dem ersten abgebrochenen Ladevorgang deutlich wird. Dies ist also schon gleich der erste (und einer der wenigen) Kritikpunkte, denn der ehrliche Anwender wird dadurch nur behindert. Außerdem – wer kommt schon mit so einem umfangreichen Programm zurecht, ohne das Handbuch gelesen zu haben? Diese Art von Kopierschutz wird sowieso ganz schnell "geknackt".

Ein Runtime-Modul, mit dem man selbstgeschriebene Programme ohne vorherige Installation von SB Professional starten kann, ist zwar im Preis inbegriffen, muß aber bei Markt & Technik angefordert werden. Danach erhält man das Modul und einen weiteren Dongle. Warum wird dieses Programm nicht gleich mitgeliefert? Will Markt & Technik seine eigenen Kunden etwa zwingen, sich registrieren zu lassen? Das sollte wohl jedem Kunden selbst überlassen bleiben.

Datamat Professional gibt sich in dieser Beziehung kundenfreundlicher. Das Programm ist in keiner Weise kopiergeschützt. Das Handbuch hat in etwa den gleichen Umfang wie das von SB Professional. Das Programm und Beispieldateien befinden sich auf insgesamt zwei Disketten. Doch zurück zu SB Professional.

Das Handbuch läßt teilweise zu wünschen übrig. Es ist für SB-Professional-Versionen auf unterschiedlichen Systemen geschrieben. Dadurch treten kleine Unstimmigkeiten auf. Dies wäre jedoch weniger schlimm, wenn das Handbuch nicht didaktisch ungünstig aufgebaut wäre. So wird zum Beispiel am Anfang des ersten Kapitels der Startvorgang von SB Professional beschrieben. Jedoch erst am Ende desselben Kapitels wird auf Besonderheiten bei unterschiedlichen Systemen eingegangen, ohne deren Kenntnis man SB Professional gar nicht starten kann.

Das Handbuch

Eine Einführung in die Bedienung des Programms, die Funktionsweise der Maus oder Eingabemöglichkeiten in Requestern wird dagegen erst am Anfang von Kapitel 3 gegeben. Auch wird man in der Einführung direkt mit theoretischem Wissen über relationale Datenbanken bombardiert, was dort absolut unangebracht ist. Es ist weiterhin fraglich, ob sich jeder Anwender, besonders die Einsteiger, mit dem Kapitel über die Programmiersprache zurechtfinden. Es sind zwar alle Befehle mit Syntax und teilweise mit kleinen Beispielen aufgeführt, aber wie man von Computersprachen weiß, reichen solche Erklärungen oft nicht aus. Hier wäre es besser, die Befehle anhand von einigen Beispielen zu erklären. Die Neuerungen der Version 3 sind in einem Addendum angefügt, was die Übersichtlichkeit nicht gerade fördert.

Das Handbuch von Datamat Professional ist im Vergleich dazu ordentlich strukturiert, für den Teil über die Programmiersprache gilt allerdings dasselbe wie für das SB-Professional-Handbuch. Im großen und ganzen kann Datamat Professional hier also erstmals ein kleines Plus verbuchen. Das Handbuch ist besser, und das Programm ist nicht kopiergeschützt.

Das erste Starten

Bei den meisten Anwendern endet das Starten nach ausgeführtem Ladevorgang mit der Fehlermeldung "Das geht nicht – Stack zu klein". Die Tatsache, daß man beim Amiga vor dem Laden (vom CLI aus) den Stack mit Hilfe des Befehls "stack 8000" auf 8000 Byte vergrößern muß, wird im

Handbuch zwar erwähnt, jedoch erst auf Seite 252 im "Anhang J" des ersten Kapitels mit dem Titel "Superbase auf dem Amiga".
Hat man SB Professional aber trotz dieser Schwierigkeiten geladen, denkt man zuerst einmal, man befände sich im Programm Superbase. Die Benutzeroberflächen ähneln sich wie ein Ei dem anderen. Das ist für Umsteiger von Superbase auf SB Professional sicherlich ein Vorteil, da sie mit den Grundfunktionen direkt vertraut sind und sich nicht erst in ein völlig neues Programm einarbeiten müssen.
Ein Blick auf die Menüleiste macht den ersten Unterschied deutlich. Der Menüpunkt "Programm" ist dazugekommen, der auch eine der zusätzlichen Fähigkeiten SB Professionals gegenüber den kleineren Versionen kennzeichnet. Das Markenzeichen von Superbase, die Iconleiste, die an die Funktionstasten eines Kassettenrecorders erinnert, ist auch bei SB Professional vorhanden. Die Kontrolle über die Datei wird über diese Tasten ausgeübt. Sie ermöglichen schnelles und langsames Durchblättern der Datei, das Suchen von Datensätzen und auch eine Stop- und eine Pausentaste sind vorhanden. Die beiden rechten Tasten haben besondere Funktionen. Die Taste, die ein Gleichheitszeichen trägt, aktiviert den Filter, eine leistungsstarke Funktion von SB Professional (zum Filter später mehr).
Die Taste, die einen stilisierten Fotoapparat darstellt, heißt "Externe Dateien" und ermöglicht es, diese auch zu zeigen. Als Möglichkeit sei hier das Anzeigen von Grafiken oder Textdateien genannt. Man kann so Grafiken archivieren, ohne diese erst in die Datei aufnehmen zu müssen. Es reicht, ein Feld mit dem Dateinamen der Grafiken anzulegen. Bei Bedarf kann man sie dann einfach durch Anklicken der beschriebenen Taste anzeigen lassen. Wer nicht gerne liest, kann sich mit dieser Taste sogar einen ASCII-Text vorlesen lassen, nicht sehr sinnvoll, aber witzig!

Variable Datenausgabe

Doch kommen wir jetzt zur Darstellung der Daten, denn was hat man von Daten, die nicht sichtbar gemacht werden können? SB Professional

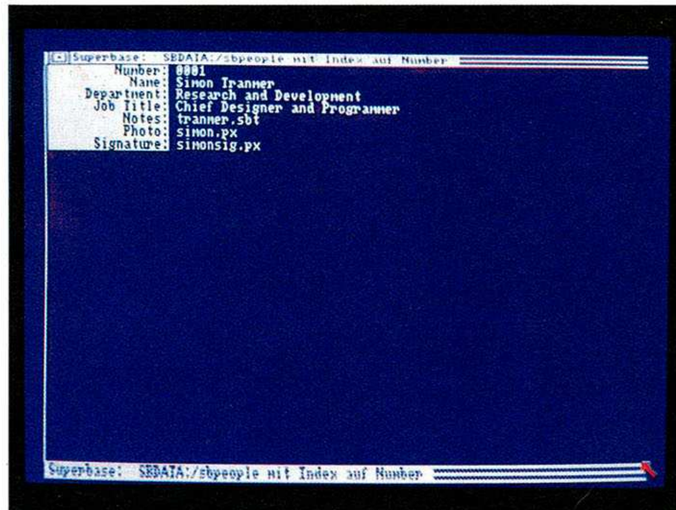


Bild 1. Wissenswertes kurz und prägnant oder...

ist in dieser Beziehung recht vielseitig. Wenn nichts anderes gewählt wird, zeigt SB Professional immer nur einen Datensatz, wobei es die einzelnen Felder untereinander auflistet. Eine Möglichkeit, um schnell Überblick über viele Datensätze zu erhalten, bietet die Tabellendarstellung. Sie listet die Datensätze untereinander auf, wobei die einzelnen Felder tabellarisch hintereinander aufgeführt werden.
Als letzte, aber vielseitigste Darstellung sei die Darstellung in Formularen erwähnt. Formulare sind selbsterstellte Masken, in denen man die Felder beliebig anordnen kann. Es ist möglich, für eine Datei beliebig viele Formulare zu erstellen. Dies ist auch sinnvoll, da Formulare meist nicht den Sinn haben, nur Daten anzuzeigen, sondern auch auszuwerten. Auf die Formulare kommen wir bei der Beschreibung des Formulareditors zurück, dort werden die Möglichkeiten ausführlicher aufgezeigt. Bei allen genannten Darstellungsarten sind die Steuerungstasten wirksam, auch in der tabellarischen Darstellung ist ein Durchblättern möglich.
Bei der Darstellung von Daten ist Datamat Professional anders ausgelegt. Datamat Professional ist eine maskenorientierte Datenbank, das heißt, daß alle Arbeiten in einer Maske geschehen. Eine Maske ist in etwa mit einem Formular in SB Professional vergleichbar. In einer Maske werden die Datenfelder frei auf dem Bildschirm angeordnet. Bei Datamat Professional wird also immer nur ein Datensatz auf einmal angezeigt. Eine Listendarstellung existiert zwar, doch ist diese recht unkomfortabel, da man in ihr nicht blättern kann; man kann sie lediglich anhalten. Allerdings kann man für die Liste wie auch für die gewöhnliche Bildschirmdarstellung in einem komfortablen Editor eine übersichtliche Maske erstellen. Im großen und ganzen ist SB Professional Datamat Professional in der Bildschirmdarstellung überlegen. Der Anwender ist einfach flexibler.

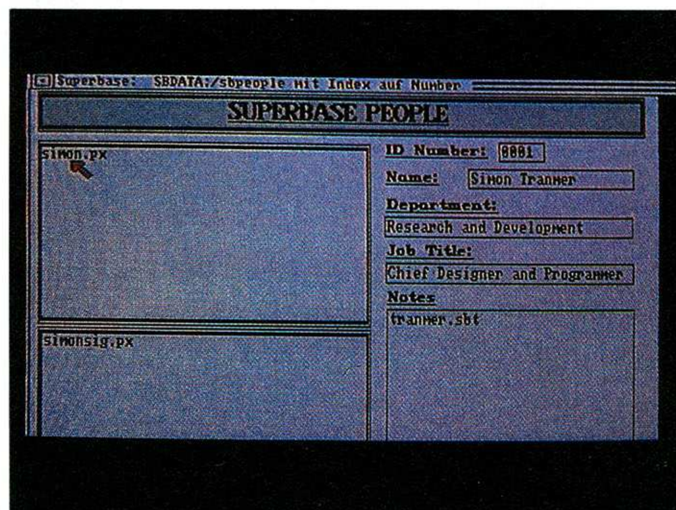


Bild 2. ...klassisch verwaltet

Erstellen von Dateien

SB Professsional unterstützt eine Vielzahl von unterschiedlichen Feldtypen. Neben Textfeldern, numerischen Feldern, Datums- und Zeitfeldern und Feldern für externe Dateien gibt es auch Berechnungsfelder, also Felder, in denen aus anderen Eingaben ein neuer Wert berechnet wird. Eine Anwendungsmöglichkeit ist zum Beispiel die Berechnung eines verzinsten Guthabens aus dem eingezahlten Betrag und dem Zinssatz. Aber auch für Textfelder können Berechnungen durchgeführt werden, zum Beispiel die automatische Erstellung eines Indexfeldes aus den ersten drei Buchstaben des Nachnamens und der Kundennummer. Die Berechnungsformel könnte dann folgendermaßen aussehen:

left$(Nachname,3) + str$
(Kundennummer)

Das zu berechnende Feld wäre dann ein Textfeld, außerdem müßten die Felder Nachname und Kundennummer (numerisches Feld) existieren. Wie man sieht, ermöglicht SB Professional auch Stringoperationen in Berechnungsfeldern. Es stehen umfangreiche Rechenoperationen zur Verfügung. Bei der Berechnung kann SB Professional auch auf Felder aus anderen Dateien zurückgreifen. Somit eröffnet sich ein großer Nutzungsbereich. Es gibt verschiedene Attribute zu Datenfeldern. Ein recht leistungsfähiges ist die Überprüfung, die auf alle Felder angewendet werden kann. Die Überprüfung läßt nur bestimmt Eingaben zu, beispielsweise kann man die Eingaben in ein Feld "Einzahlung" durch die Überprüfungsformel "Einzahlung > 50" auf solche beschränken, die größer als 50 sind. Auch Texte können überprüft werden.
Berechnungs- und Überprüfungsfelder bietet im gleichen Umfang auch Datamat Profes-

sional, hier steht es Superbase Professional nicht nach. Nach abgeschlossener Felddefinition besteht die Möglichkeit, Indizes zu definieren. Möglich sind maximal 999, ein Index muß immer definiert sein. Es wird dabei zwischen Einfachindex und Mehrfachindex unterschieden. Wählt man die Erstellung eines Einfachindex für ein Feld, kann keine Eingabe doppelt vorkommen. Sinnvoll ist dies bei Feldern für Kunden- oder Seriennummern. Bei der Indizierung von Namensfeldern wählt man dagegen besser einen Mehrfachindex, denn es gibt Namen, die häufig vorkommen, wie zum Beispiel "Müller". Diese Möglichkeit bietet Datamat Professional nicht.

Über Felder derselben Definition lassen sich verschiedene Dateien verknüpfen. Ein Beispiel ist ein Feld "Kundennummer". Wenn in mehreren Dateien gleich definierte Felder vorkommen, kann man sie über dieses Feld verbinden. Man kann dann mit der Eingabe der Kundennummer Daten aus verschiedenen Dateien abrufen. Es wird so ermöglicht, zum Beispiel eine Adressendatei und eine Guthabendatei über die Kundennummer zu verbinden. Auf diese Weise müssen Daten unterschiedlicher Art nicht in einer Datei gespeichert werden. Eine solche Art der Verbindung von Dateien heißt relationale Datenverwaltung.

Arbeiten mit SB Professional

Die Arbeit gestaltet sich sehr variabel. Es ist möglich, beliebig viele Dateien gleichzeitig zu öffnen. Man kann von der Benutzeroberfläche aus beliebige Felder schließen, diese werden dann einfach nicht mehr angezeigt. Daneben ist es auf einfache Art und Weise möglich, die Länge einzelner Felder zu verändern, Felder hinzuzufügen oder zu entfernen.

An dieser Stelle ist eine Beschreibung der Filterfunktion sinnvoll. Der Filter bewirkt die Darstellung ausgewählter Datensätze. Der Anwender kann ohne große Umstände Formeln definieren, die bewirken, daß nur noch solche Datensätze erscheinen, die die in der Formel gestellte Bedingung erfüllen. Durch die Formel "Einkommen > 5000" werden bis zum Ausschalten des Filters nur noch solche Datensätze angezeigt, bei denen im Feld "Einkommen" eine Zahl größer als 5000 steht. Auch Textfelder lassen sich überprüfen. Ein fiktives Anwendungsbeispiel sei dazu genannt: Das Versandhaus "Werner & Co" möchte zum zehnjährigen Jubiläum allen Kunden, die Werner heißen oder zu den ersten 200 Kunden des Unternehmens zählen oder aufgrund häufiger Bestellungen zu den sehr guten Kunden des Unternehmens gehören, ein kleines Präsent schicken.

Dies ist mit dem Filter kein Problem. Zuerst erstellt man die Formel mit Hilfe des Kommunikationsfensters. Sie lautet folgendermaßen:

Nachname LIKE "Werner" OR Kundennummer < = 200 OR Status = "Wichtig"

Anschließend aktiviert man den Etikettendruck, und schon druckt SB Professional die Adressenaufkleber. Auch einfache Serienbriefe kann man mit SB Professional schreiben. Das Programm beinhaltet zu diesem Zweck einen einfachen Texteditor. Dieser bietet allerdings nur wenige Mittel zur Texterstellung, Ausrichtung des Textes ist überhaupt nicht möglich. In den Text werden die Namen der Datenfelder oder – dies ist eine Verbesserung der Version 3 – auch Variablen eingefügt, die dann für bestimmte Daten stehen.

In der Version 3 kann man Daten aus anderen Programmen direkt einlesen. Dazu zählen zum Beispiel dBase-Dateien oder Lotus-Dateien. Weiterhin besteht die Möglichkeit, ASCII-Dateien einzulesen. Da fast alle Programme ASCII-Dateien schreiben können, stellt der Datenaustausch somit keine Probleme dar. Mit der Export- und der Import-Funk-

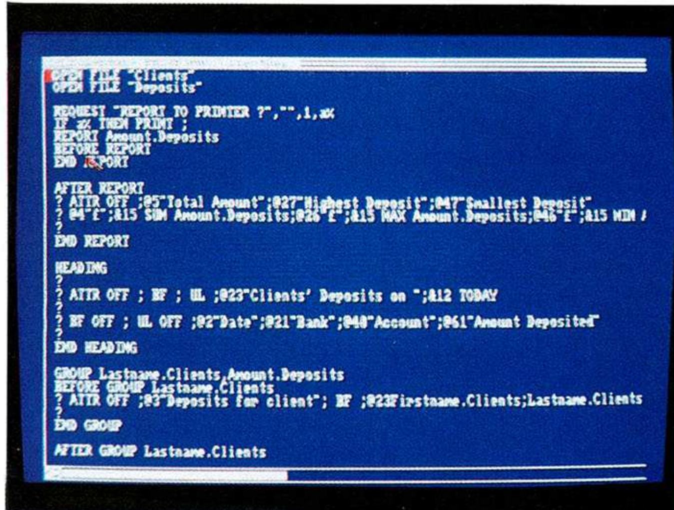


Bild 3. Programme mit DML erstellt

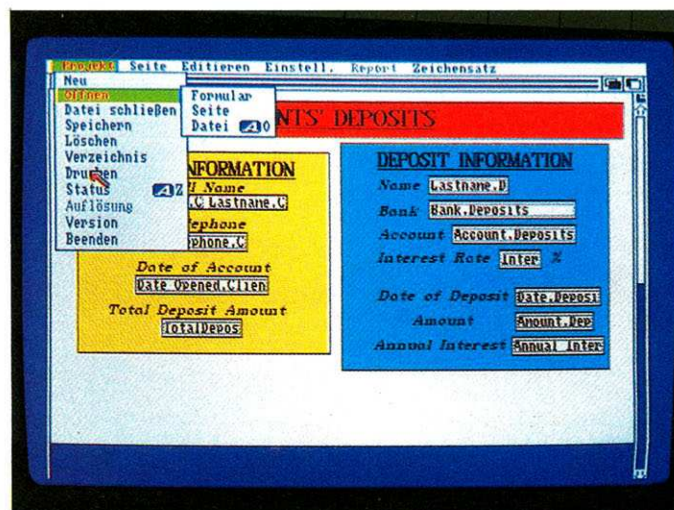


Bild 4. Der Formulareditor ist ein mächtiges und hilfreiches Feature

Kleinanzeigen

Biete Software

Brunosoft/Schreiber
PD-Software für den Amiga
Katalog kostenlos
Bei: Brunosoft/Schreiber,
1000 Berlin 51, Sommerstr. 37 **G**

CH-Severex. Wir bieten Dir die aktuellste Soft, die es auf dem Markt gibt. Severex, Postbox 207, CH-9425 Thal, BBS: 071/71 54 10

Verk. Aztec C V3.6a Developer ab 18 Uhr, 05 41/1 79 81

PD-Soft für den Amiga
Katalogdisk kostenlos
Hardware-Virusprotector 30,– DM
Best. an: Brunosoft/Schreiber,
1000 Berlin 51, Sommerstr. 37 **G**

Digitalisierte Erotik-Diaschau zu verkaufen. Hohe Auflösung – 4096 Farben. Info gegen Freiumschlag bei: Thomas Altmüller, Turpinstr. 29, 5100 Aachen **G**

Club

Neuer Computerclub in Neuss INTERESSENGEMEINSCHAFT COMPUTER NEUSS (ICN). Die Mitglieder treffen sich jeden zweiten und vierten Samstag im Monat, der Clubbeitrag beträgt 10,– DM monatlich. Neue Mitglieder (ab 16 Jahre), auch Computerneulinge, sind herzlich willkommen. Kontakt: R. Ackermann, Fesserstr. 10, 4040 Neuss 1, Tel. (0 21 01) 54 61 68

tion können auch Daten zwischen einzelnen SB-Professional-Dateien ausgetauscht werden.

Eine weitere und wichtige Funktion von SB Professional ist die Abfrage-Funktion, mit der sich Daten auswerten lassen. So können beispielsweise Datensatzzähler hinzugefügt und Gesamtsummen angezeigt werden. Auch kann man mit dieser Funktion bestimmte Felder schließen, mit einem Filter nur gewisse Datensätze zulassen, die Art der Sortierung bestimmen und das Layout steuern. Ein sehr umfangreiches Feature ist das Erstellen von Reporten. Aus den ausgewählten Datensätzen kann man leicht Mittelwerte und Gesamtsummen berechnen und die Datensätze zählen lassen.

Immer hübsch up to date sein

Ebenfalls hilfreich ist die Aktualisierungsfunktion. Wird sie aufgerufen, werden alle Berechnungen neu durchgeführt. Sie ist nötig, da Berechnungen auch auf Werte zurückgreifen können, die einer verknüpften Datei entstammen. Wird in einer Datei ein Wert geändert, stimmen die berechneten Werte in der verknüpften Datei nicht mehr. Die Funktion "Aktualisieren" berechnet deshalb die Werte in allen relational verbundenen Dateien neu. SB Professional bietet in der Version 3 ebenfalls die Unterstützung der seriellen Schnittstelle, das heißt, Datenaustausch über Modem, Akustikkoppler oder Nullmodem stellt kein Problem dar. Mit SB Professional können die Übertragungsparameter in weiten Grenzen beeinflußt werden.

Datamat Professional stellt ebenfalls fast alle hier aufgeführten Funktionen zur Verfügung, es fehlt jedoch die Möglichkeit, Reporte zu erstellen oder Daten komfortabel über die serielle Schnittstelle auszutauschen. Mit dem Fehlen einer Report-Funktion mangelt es Datamat Professional an einer vielseitigen Funktion zur Datenauswertung. Um in Datamat Professional solche Auswertungen zu erstellen, ist es unumgänglich, die Programmiersprache zu Hilfe zu nehmen.

Der Formulareditor ist ein wichtiges Modul. Er liegt als eigenständiges Programm vor und öffnet einen eigenen Screen. Damit kann man Formulare erstellen, die unterschiedliche Funktionen haben können. Zum einen kann man dieses Feature zur optisch ansprechenden Präsentation von Datensätzen nutzen, zum anderen aber auch zur Darstellung von Ausgabeformularen für Reporte. In diesen Ausgabeformularen kann man Berechnungen anstellen, außerdem kann man Daten aus verschiedenen Dateien übernehmen. Weiterhin dienen erstellte Formulare zur Druckausgabe; Rechnungen oder ähnliches können auf diese Art und Weise ansprechend ausgedruckt werden.

Wo man hinsieht, überall Formulare

Der Formulareditor arbeitet objektorientiert. Zur grafischen Gestaltung stellt er Rechtecke, Flächen, Textfelder und Linien zur Verfügung. Es können auch Grafiken aus Malprogrammen in die Maske eingefügt werden. Wenn man genügend Speicherplatz zur Verfügung hat, kann man Formulare in Auflösungen bis hin zum Interlace-Modus und mit bis zu 16 Farben arbeiten. 1 MByte Hauptspeicher reicht jedoch kaum aus. Zum Füllen der Flächen stehen einige Muster zur Verfügung. Die unterschiedlichen Objekte, die der Editor bietet, haben eine festgelegte Priorität, die man nicht ändern kann.
Wie bereits erwähnt, kann man die Formulare auch für die Erstellung von Berichten verwenden, die ein ansehnliches Äußeres aufweisen sollen. Das Spektrum ist hier sehr vielfältig. Der Formulareditor ist ein leistungsfähiges Tool, das man nach kurzer Einarbeitungszeit nicht mehr missen möchte. Datamat Professional arbeitet etwas anders, es besitzt drei Maskeneditoren, einen für die Druckmaske, einen für die Listenmaske und einen für die Ausgabemaske. Die Editoren sind sich sehr ähnlich und zudem bezüglich der Grafikmöglichkeiten umfangreicher als der Formulareditor von SB Professional. Allerdings erfüllt die Bildschirmmaske bei Datamat Professional einen anderen Zweck. Das Programm arbeitet maskenorientiert, das heißt, daß zu jeder Datei mindestens eine Bildschirmmaske existiert, die einzig und allein ihre Verwendung darin findet, die Daten anzuzeigen.
Ein wichtiges Element von SB Professional ist die Programmiersprache. Sie ist in das Programm integriert. Mit dem Menüpunkt "Programm" aus der Menüleiste kann man alle benötigten Funktionen bedienen. SB Professional hat einen eigenen Editor eingebaut, mit dem man seine Programme schreiben kann. Die Programmiersprache selbst ist stark an BASIC angelehnt. Sie bietet alle Funktionen, die auch über die Menüs aufgerufen werden können. Zusätzlich kann man sich seine eigene Benutzeroberfläche schaffen und eigene benötigte Menüs und Requester erstellen.

DML – ran ans Programmieren

Ansonsten kann man mit der Programmiersprache häufig benötigte Anwendungen schreiben, die dann durch Starten des Programms automatisch ablaufen. Ferner besteht die Möglichkeit, ein Programm nach dem Laden von SB Professional automatisch starten zu lassen. Dieses Programm kann dann zum Beispiel schon selbsttätig die benötigten Dateien öffnen, Formulare laden, den Filter definieren und Reporte erstellen.

Datamat Professional besitzt ebenfalls eine eigene Programmiersprache, PROFIL genannt. Sie hat mit DML einiges gemeinsam. Auch sie lehnt sich eng an BASIC an. Sie ermöglicht dem Anwender vor allem, Daten in einem gewissen Rahmen auszuwerten. Man kann sagen, daß die Programmiersprachen in etwa gleichzusetzen sind. Ein großer Nachteil von Datamat Professional liegt allerdings darin, daß es nicht möglich ist, ein Programm unabhängig vom Hauptprogramm ablaufen zu lassen, wie es bei Superbase durch das Runtime-Modul möglich ist.

Darf's ein wenig mehr sein?

Die Frage, die sich hier vornehmlich stellt, lautet: Nimmt man ein günstiges Programmpaket oder zieht man eher das teurere vor? Diese Frage muß jeder für sich selbst beantworten. Datamat Professional ist zwar um 200 DM günstiger, entbehrt dafür aber auch einiger Funktionen, die SB Professional sehr leistungsfähig machen. Auch ist das Programm sehr verspielt. Wer auf die zusätzlichen Funktionen verzichten kann, ist mit Datamat Professional besser bedient, zumal er ein besseres Handbuch in die Hand bekommt und nicht mit dem Dongle kämpfen muß.

Wer allerdings auf die zusätzlichen Funktionen von SB Professional nicht verzichten möchte und sich durch das teilweise schlechte Handbuch nicht abschrecken läßt, ist mit diesem Programm auch sehr gut bedient. Allerdings muß hier ein höherer Preis in Kauf genommen werden. Wer seine selbstgeschriebenen Programme verkaufen möchte, sollte auch auf die Lizenzgebühren achten. Die Lizenzgebühr für ein weiteres Runtime-Modul beträgt 30 DM, eine Voll-Lizenz kostet 5000 DM, das ist eine ganz schöne Stange Geld. Datamat Professional bietet diese Möglichkeiten nicht. Möchte man ein selbstgeschriebenes Programm verkaufen, so muß der Käufer immer im Besitz von Datamat Professional sein. Ein Runtime-Modul gibt es nicht.

Und hier noch ein Tip für die Besitzer von Superbase Professional: Wer mit dem Handbuch nicht zurechtkommt, der sollte sich einmal "Das große Buch zu Superbase" aus dem Data-Becker-Verlag anschauen. Hier werden alle Funktionen von Superbase und die Programmiersprache selbst anhand von Beispielen anschaulich erklärt.

(vb)

AMIGA DOS Blitzlicht

Name: Superbase Professional Version 3
Preis: 599,– DM
Anbieter: Markt & Technik Hans-Pinsel-Str. 2 8013 Haar

Positiv:
- äußerst flexibles und leistungsfähiges Programm
- separate Runtime-Module
- zur Zeit beste Dateiverwaltung

Negativ:
- Kopierschutz (Dongle)
- didaktisch schlechtes Handbuch

Andreas Polk

Fancy3DFonts

Animators linke Hand

Egal, ob Textverarbeitung oder Animationsprogramm, ein gerüttelt Maß an unterschiedlichen Fonts geben erst das gewisse Etwas.

Eine sehr leistungsfähige, jedoch nicht sehr beachtete Funktion von Sculpt-4D ist das Einbinden eigener Fonts. Das liegt daran, daß mit dem Programm nicht ein einziger Zeichensatz ausgeliefert wird. Um diese hervorragende Funktion zu nutzen, muß man entweder erst einen Font selbst erzeugen, was sehr viel Zeit in Anspruch nehmen kann, oder man kauft sich entsprechende Fontdisketten.

Unter dem Namen Fancy3D-Fonts werden für TurboSilver 3.0 und Sculpt-4D drei dieser Zeichensätze verkauft. Den Vertrieb hierfür hat die Firma Heinrichson, Schneider & Young übernommen. Wir haben für Sie die Fontdisketten für das Programm Sculpt-4D angeschaut und möchten in diesem Bericht eigene Erfahrungen wiedergeben. Dabei wird nicht nur auf die Fonts selbst eingegangen, sondern auch ein wenig das Umfeld beleuchtet.

Einbinden von Fonts

Möchte man mit einem Animationsprogramm Schriften erzeugen, so muß man gewöhnlich jeden einzelnen Buchstaben als Objekt einladen und plazieren. Jeder, der das schon einmal gemacht hat, weiß, wie mühsam solch eine Arbeit sein kann. Sculpt-4D geht hier einen anderen Weg. Mit diesem Programm ist es möglich, einen Font über die Tastatur einzugeben, 'Text' also richtig zu schreiben. Gerade in Verbindung mit den Fancy3DFonts macht sich dieses Feature besonders gut.

Zunächst muß der Anwender festlegen, welchen Font er benutzen möchte. Dazu muß der Font erst einmal geladen werden. Sculpt-4D erlaubt nicht, mehrere Fonts gleichzeitig im Speicher zu verwalten. Hat man einen Font geladen, so wird nun bei jedem Tastendruck in dem entsprechendem Verzeichnis nach einem Objekt mit einem Namen gesucht, der durch die gedrückte Taste genau bestimmt ist. So wird beim Drücken der Taste 'a' beispielsweise nach dem Objekt (denn ein Sculpt-Font besteht ja letztendlich aus Objekten) 'a__3D.scene' gesucht. Die genaue Namensgebung kann man im Handbuch selbst nachlesen.

Das Zeichen wird eingeladen und der Cursor auf die nächste Position gestellt. So kann man den Text Zeichen für Zeichen eingeben. Es ist sogar möglich, im Schreibmaschinentempo zu schreiben, da Sculpt-4D die Eingaben in einen Puffer lagert. So geht kein Zeichen verloren. Wird einmal ein Objekt nicht gefunden, so blinkt der Bildschirm auf und die nächsten Objekte werden eingeladen, falls noch Tasten gedrückt wurden. Kommen wir nun zu den Fancy3DFonts selbst. Sie werden auf zwei Disketten ausgeliefert. Ein Handbuch ist überflüssig. Lediglich ein Beipackzettel und ReadMe-Files geben Erklärungen. Allerdings wäre ein Handbuch hier auch ziemlich überflüssig, denn was sollte schon erklärt werden?

Auf der ersten Diskette befindet sich der komplexeste der drei Zeichensätze. Es handelt sich hierbei um eine Times-ähnliche Schrift. Auf der anderen Diskette befindet sich eine Kursivschrift und eine Fettschrift. Der erste Eindruck, den man erhält, ist sehr gut. Die Buchstaben sind sauber erstellt worden und der Cursor wird genau an die richtige Position gesetzt, so daß man flüssig tippen kann. Neben den gängigen Buchstaben befinden sich noch Umlaute, französische Zeichen, Zahlen und einige Sonderzeichen auf den Disketten. Diese Objekte müssen teilweise aber noch umbenannt werden, da Sculpt-4D beim Drücken der Taste 'ß' nie auf die Idee käme, nach einem Objekt mit dem Namen 'GermanS__3D.scene' zu suchen. Das Umbenennen ist schnell erledigt und nicht weiter schwierig.

Alle Schriften werden in weißer Farbe mit matter Oberfläche (dull) ausgeliefert. Der Anwender kann das so belassen oder muß die Materialeigenschaften nach seinen Wünschen ändern. Der einzige Nachteil ist der, daß die Ergänzungen zu den gängigen Tasten unvollständig sind. So enthalten zwei Zeichensätze zwar das 'ß', nicht aber der dritte. Genauso fehlen bei allen Zeichensätzen die Sonderzeichen '" ▲ % / ( ) = -'. Da die Fancy3D-Fonts nicht gerade preiswert sind (knapp 150,- DM darf man hinblättern), dürfte so etwas nicht vorkommen.

Jeder, der einmal mit Sculpt-4D und den Fancy3D-Fonts gearbeitet hat, wird diese einfache Art, Text eingeben zu können, nicht mehr missen wollen. Das einzige, was Anwender von Sculpt-4D an der Anschaffung dieser Fonts hindern könnte, ist der hohe Preis.

(vb)

AMIGA DOS Blitzlicht

Name: Fancy3DFonts
Preis: 148,– DM
Hersteller: Heinrichson, Schneider & Young
Classen-Kappelmann-Str. 24
5000 Köln 41
Tel.: 0221/402365

Positiv:
- gute Qualität
- viele Buchstaben und Sonderzeichen
- kein Kopierschutz

Negativ:
- einzelne Zeichen fehlen
- hoher Preis

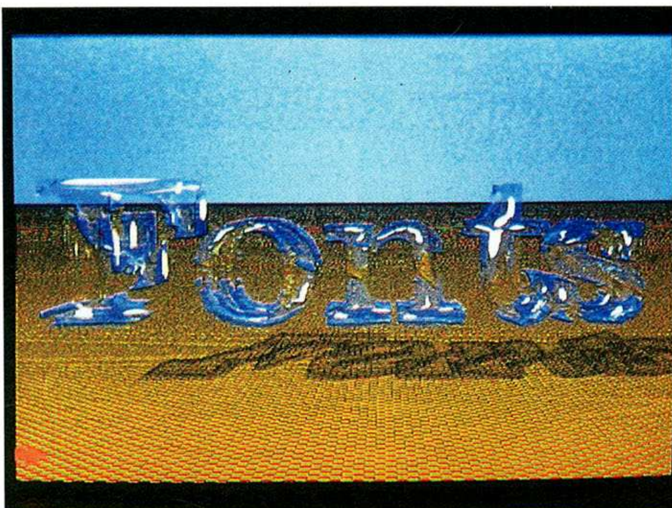


Bild 1. Fancy Fonts im Raytracing-Format



Bild 2. Fancy Font Ital mit Sculpt-4D berechnet

Scan me up, Scottie!

Mit Scannern vom Bild zur Computergrafik

Unser Titel mag auf den ersten Blick etwas seltsam erscheinen, wurde er doch ansatzweise einer bekannten Science-fiction-Serie entnommen. Im Grunde jedoch beschreibt er kurz und prägnant, worum es geht: lebende Personen bildtechnisch in Computergrafiken umzuwandeln. Ein Vorgang, der sich zwar nicht mit dem 'Beamstrahl' erreichen läßt (den gibt's noch nicht), aber mit Scannern – und die gibt's! Zum Beispiel von der Firma Reisware.

Seit man gemerkt hat, daß sich Computer ganz hervorragend zum Erstellen von Printmedien eignen, ging die Überlegung um, wie man den gestreßten Anwendern dieser Nachfolger der Textverarbeitungen das Einbinden von Grafiken in die Layoutseiten erleichtern könnte. Nicht jeder Möchtegern-Zeitungsverleger ist auch begnadeter Grafiker, und so endeten die meisten Versuche dieser Art mit einer recht anspruchslosen Strichmännchengrafik. Im Ernst, stellen Sie sich die Amiga-DTP-Anlage vor, auf der intelligente und flott geschriebene Texte mit handgekritzelten Zeichnungen unterlegt werden. Das kann nicht der Sinn des Ganzen sein.

Findige Tüftler kamen nun auf die Idee, Vorlagen mit technischen Hilfsmitteln in digitale Signale umzuwandeln und in die entsprechenden Computer einzulesen. Keine schlechte Idee, sie enthielt das Grundprinzip der heutigen Bildlesegeräte, jedoch mit sehr vielen Einschränkungen: Die ersten Geräte konnten nur zwischen hell und dunkel unterscheiden und daher auch nur die entsprechenden Bits auf 0 oder 1 setzen. Also nicht gerade das Schwarzweiße vom Ei, wie man jetzt treffend bemerken könnte.

Mit fortschreitender Technik wurden allerdings nicht nur die Computer verbessert, sondern auch die Peripherie-Geräte – so auch die Scanner. Sie konnten jetzt Bilder in Graustufen umwandeln und diese Informationen weitergeben. Die eingelesenen Bilder wurden schärfer auf dem Monitor dargestellt und an bestehende Grafiknormen (zum Beispiel IFF) angepaßt, ließen sich auch unter den entsprechenden Grafikprogrammen weiterbearbeiten. Die Preise waren allerdings noch horrend, Privatleute hätten schon Lottogewinner sein müssen, um einen solchen Scanner bezahlen zu können. Bald danach tauchten die ersten Farbscanner auf, die Farbbilder in (fast) originalgetreuen Farben wiedergeben konnten. Außerdem wurden die Geräte kleiner, sie konnten mit einer Hand bedient werden – ein neuer Begriff wurde kreiert: der Hand-Scanner. Und damit sind wir beim eigentlichen Thema: die Handy-Scanner der Firma Cameron.

Beide Scanner werden in stabilen Verpackungen geliefert. Bei unseren Testversionen lag neben dem Scanner, dem Interface und der Software noch je ein deutsches und ein englisches Handbuch bei sowie einige Hinweise zur Installation des Gerätes.

Für jeden Amiga der richtige Anschluß

Diese ist recht schnell erledigt. Da das Interface am Prozessorbus angeschlossen wird, braucht man am Amiga 500 und Amiga 1000 nur die kleine Klappe an der linken Seite zu entfernen (beim Amiga 1000 an der entsprechenden Stelle) und das Interface anzuschließen. Der Scanner wird über einen neunpoligen Stecker, ähnlich wie die Maus, an das Interface angeschlossen. Wird jetzt noch die Software gestartet, kann der Spaß losgehen.

Zum Betrieb am A2000 ist ein bißchen mehr Aufwand nötig. Hier benötigt man zuerst eine Adapterplatine, die den DMA-Slot des A2000 in einen CPU-Port verwandelt. Diese kleine Zusatzplatine wird in einen freien Slot des A2000 gesteckt, auf diese dann das Interface. Zusätzlich befindet sich noch eine Extrabuchse, die auf einer der Slot-Blenden montiert ist, mit einem Verlängerungskabel in der Lieferung. Die Blende mit der Buchse wird nun anstelle einer Blindblende hinten eingebaut und das Verlängerungskabel an das Interface angeschlossen. Hier hat man wieder einen Vorteil des A2000 vor Augen: Das Interface befindet sich unsichtbar innerhalb des Rechners, und der Scanner kann direkt an eine Buchse angeschlossen werden, die fest am Gehäuse angebracht ist, beim A500 beziehungsweise am A1000 hingegen muß man das Interface als 'Rucksack' mitnehmen. Allerdings kann es beim A2000 eventuell zu Platzproblemen kommen: Das Testgerät, ein A2500 mit RAM-Karte (Slot 1) und Autoboot-Filecard (Slot 2) mit externem Controller (Slot 3), sollte jetzt auch noch das Scanner-Interface verkraften. Wir wollen niemand auf die Folter spannen: Es ging, zwar ein bißchen gedrückt, aber es ging. Hier muß man den Entwicklern des Interfaces Hochachtung zollen, anscheinend haben sie ihre Entwicklung tatsächlich in solchen 'Härtetests' (randvoller Amiga) ausprobiert. Für eine PC-Karte war nun jedoch leider kein Platz mehr, aber was soll's.



Bild 1. Die beiden getesteten Scanner: Typ 10 (Schwarzweiß) und Typ 6 (Color)

Der Einbau ist fast 'kinderleicht'; die Installation wird genau erklärt, und zur Not befinden sich immer noch Hinweise auf den Platinen.

Die Scanner selbst sind recht klein und liegen gut in der Hand. Außerdem sind sie leicht und gut zu führen, die Rolle garantiert eine störungsfreien Lauf. Sie ist übrigens das einzige Teil, bei dem ab und zu auf Sauberkeit geachtet werden muß, da sonst die Impulsfolgen nicht mehr gleichmäßig an die Elektronik gegeben werden können, der Scanner also nicht mehr richtig scannt. Der Scan-Vorgang beginnt erst mit Drücken einer an der Oberfläche befindlichen Taste, ein genaues Ausrichten ist also bei den Geräten möglich. Dieses Ausrichten ist übrigens bei den beiden Typen unterschiedlich, wir gehen später noch näher darauf ein. Das Verbindungskabel vom Interface zum Scanner ist ausreichend lang (zirka 2m), man wird also nicht irgendwann den Scan-Vorgang abbrechen müssen, weil der Stecker aus der Buchse gezogen wurde. Doch nun zu den einzelnen Typen.

Der Colorscanner – Farbbilder in verschiedenen Modi

Der Colorscanner Typ 6 ist ein Gerät zur Übermittlung von Farbvorlagen, die von den gängigen Grafikprogrammen weiterbearbeitet werden können. Auf der Systemdiskette befinden sich mehrere Programme, die das Arbeiten mit dem Scanner erlauben.



Bild 2. Mit Zeichenprogrammen können die Bildvorlagen verändert oder verbessert werden

○ Hier wäre zuerst einmal das Programm 'Demo' zu nennen. Es ist als erster Test zu gebrauchen, um festzustellen, ob alle Anschlüsse richtig angebracht wurden. Nach dem Start des Demo-Programms beginnt die LED-Zeile des Scanners zu leuchten; ein Druck auf die Taste und der Scan-Vorgang beginnt. Das Programm läßt nur einen Schwarzweiß-Scan zu und wird durch Mausklick wieder beendet. Wie gesagt, es kann nur einer ersten Orientierung dienen, ob alles so funktioniert, wie es soll. Man sollte deshalb nicht erschrecken, wenn man ein Bild einscannt und statt dessen einen Schattenriß bekommt. Der erste Test sollte mit einem Text beginnen, also Scanner kurz über das Handbuch rollen und auf den Bildschirm achten. Ist der Text gut lesbar, ist alles in Ordnung.

○ Das Programm 'Colorscanner' dient der direkten Übernahme von Farbvorlagen in die Bitmap des Amiga. Hier muß gleich eines noch gesagt werden: Scanner benötigen einiges an Speicherplatz; mindestens 1 MByte wird benötigt, um den Scan-Betrieb aufrecht zu erhalten. Allerdings gilt hier die Maxime: Je mehr Speicher, um so besser.

'Colorscanner' wird, wie beim Amiga üblich, über ein Menü gesteuert, welches mehrere Funktionen enthält, die allesamt mit der Maus anklickbar sind. Im 'Project'-Teil findet man außer einer kurzen Info zum Programm den Startteil des Scan-Vorganges. Hierbei wird ein Bild vom oberen bis zum unteren Rand des Bildschirms eingelesen. Daneben gibt es noch die obligatorische Lade- und Speicherroutine, wobei entweder das komplette Bild oder ein markierter Ausschnitt als Brush (Pinsel) gespeichert wird. Eine implemtierte Druckroutine druckt das Bild sofort auf einen unter Preferences eingestellten Drucker aus.

Im Menüteil 'Edit' kann man einen bestimmten Auschnitte des Bildes markieren, ausschneiden, kopieren oder einfügen. Eine 'Undo'- und 'Clear'- Funktion sind ebenso enthalten wie eine 'Palette'-Funktion zum Verändern der Farben.

Im 'Display'-Menü geht es schließlich darum, in welchem Modus unser Bild aufgenommen werden soll. Hier hat man mehrere Wahlmöglichkeiten:

- den 16-Farben-Modus,
- den 32-Farben-Modus,
- den Extra-Halfbrite-Modus mit 64 Farben,
- und den HAM-Modus mit 4096 Farben

Außerdem kann man noch folgendes bestimmen:

- normale Darstellung unter Zuhilfenahme der Standardpalette,
- optimierte Darstellung mit verbesserter Farbpalette
- Gamma-Korrektur ein- und ausschalten,
- Wiedergabe mit oben eingestellten Parametern.

Der Scan-Vorgang wird zeitgleich auf dem Bildschirm gezeigt, dazu wird das Bild in ein achtstufiges Grauraster aufgeteilt. Erst nach Abschluß des Scan-Vorgangs wird das Bild mit der Farbpalette erstellt. Manche Bildvorlagen wurden im 32-Farben-Modus ganz gut nachgebildet, die besten Ergebnisse erzielt man jedoch im HAM-Modus (Hold And Modify), da hier das Farbwunder Amiga auf seine komplette Farbpalette von 4096 Farben zurückgreifen kann.

Die Gamma-Korrektur wird angewandt, wenn die Farben der Vorlage sehr dunkel sind. Durch die Korrektur wird über eine Aufhellung eine kontrastreichere Wiedergabe erreicht. Die Farbpalette wird wie bei bekannten Zeichenprogrammen zum Einstellen der Farben verwendet.

Das Programm 'Colorscanner' arbeitet mit zwei Seiten. Auf der zweiten Seite lassen sich direkt Ausschnitte der ersten Seite einsetzen. Auf diese Weise lassen sich schnell und einfach Montagen aus gescannten Vorlagen zusammenbasteln. Die Montage kann allerdings nur im 'Normal'-Modus geschehen; sollten die Ausschnitt-Bilder in Falschfarben erscheinen, lassen sie sich immer noch mit der 'Redraw'-Funktion in einen anderen Modus umwandeln.

Um Bilder vom Schreibtisch in den Amiga zu bringen und dazu noch per Grafikprogramm weiterzubearbeiten, reicht das 'Colorscanner'-Programm völlig aus. Die Vorlagen werden in einer hochwertigen Qualität übernommen und können jederzeit in weiterverarbeitende Programme übernommen werden. Ein Versuch mit Professional Page, dem DTP-Programm, bewies, daß auch hier die Bildvorlagen in die entsprechenden Seiten-Layouts ohne Probleme eingebunden werden konnten.

○ 'Handy-Painter' ist ein Zeichen- und Bildbearbeitungsprogramm, daß sich hinter anderen Programmen nicht zu verstecken braucht. Hier können die digitalisierten Vorlagen überarbeitet und mit grafischen Einlagen oder Text versehen werden. Das Programm enthält eine Vielzahl an Funktionen, die wir hier aus Platzgründen nicht alle aufzählen können. Nur soviel: Eigene Font-Dateien erlauben die Auswahl unter einer Vielzahl an Schriftarten; Muster zur Füllung von Flächen können frei definiert werden. Dem Programm stehen acht virtuelle Bildspeicher zur Verfügung, auf denen separat gearbeitet werden kann. 'Handy-Painter' ist an sich schon ein Layout-Programm, das zur Erstellung und Ausgabe von Layout-Seiten unter Zuhilfenahme von Scanner-Vorlagen genutzt werden kann. Das Programm ist ausgeklügelt und funktionsreich, Besitzer des Scanners sind also nicht angewiesen, sich sofort noch bildverarbeitende Software dazuzukaufen.

○ 'Scan & Save' arbeitet ähnlich wie 'Colorscanner', jedoch mit Schwarzweißgrafiken. Im Programm kann man die maximale Anzahl der Zeilen, die Zeilenlänge und den Dateinamen angeben, unter dem das gescannte Vorbild abgespeichert werden soll. Das Programm legt ein Unterverzeichnis namens 'pictures' an, in dem sich die vom Scan-

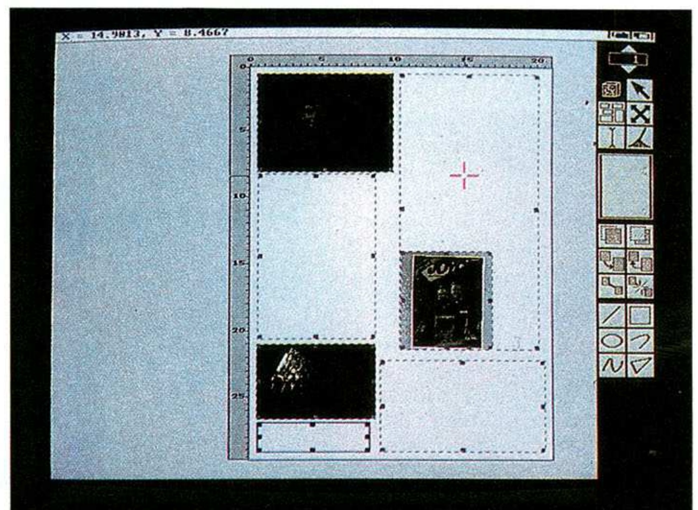


Bild 3. Bildvorlagen können nach dem Scannen natürlich in den entsprechenden Programmen weiterbearbeitet werden, hier ein Beispiel unter Professional Page

ner eingelesenen Bilddateien wiederfinden. Der Scan-Vorgang wird nach Erreichen der letzten angegebenen Zeile automatisch abgebrochen.

Wer nicht tippen will, soll scannen!

Der Handy-Scanner Typ 10 ist das preiswertere Schwarzweißmodell. Preiswerter bedeutet hier aber nicht weniger, im Gegenteil. Zwar kann man mit diesem Scanner keine Farbvorlagen erzeugen, dafür wird jedoch eine weitere Software mitgeliefert – das 'Reader'-Programm.

Die Software an sich ist die gleiche wie beim Farbscanner, das Demo-Programm, der Handy-Painter und 'Scan & Save' gehören hier ebenfalls zur Ausstattung.

Neu ist, wie schon gesagt, der 'Reader', also der Leser. Und hier sind wir schon bei einer interessanten Verwendung des Schwarzweiß-Scanners: dem Einlesen von Textdateien. 'Reader' ist ebenfalls ein über Menü und Maus gesteuertes Programm. Per Scanner können Texte eingelesen werden, die in der maximalen Auflösung und der Länge eingestellt werden können. Dank der mitgelieferten Zeichen-Datensätze kann nun eine Erkennung des gescannten Textes gestartet werden, indem das Programm den Scan-Text in einem Grafikfenster darstellt und der Scanner-Benutzer anhand eines Textfensters das Ergebnis zu sehen bekommt.

Außerdem enthält 'Reader' einen einfachen Texteditor ähnlich einer sehr einfach gehaltenen Textverarbeitung. Der Clou ist jedoch, daß hier nicht nur die Tastatur, sondern auch der Scanner als Eingabeteil genutzt wird.

Wenn das Scannen von Bildvorlagen ein Kinderspiel war, so beginnt hier die Arbeit. Texterfassung will gelernt sein, es gehört eine Portion Übung dazu, einen ersten Text in ASCII-Form umzuwandeln. Wichtig ist, daß man zuerst einmal den Zeichensatz bestimmt, in dem die Vorlage geschrieben wurde und diesen als Datensatz lädt (wobei mehrere Datensätze geladen werden können, um die Wiedererkennung zu erleichtern). Das Programm muß mit etwas Vorsicht behandelt werden, allzu schnelle Wechsel von Funktionen oder Ungeduld beim Arbeiten mit dem Scanner kann unbefriedigend enden, wenn die Texte partout nicht umgewandelt werden wollen.

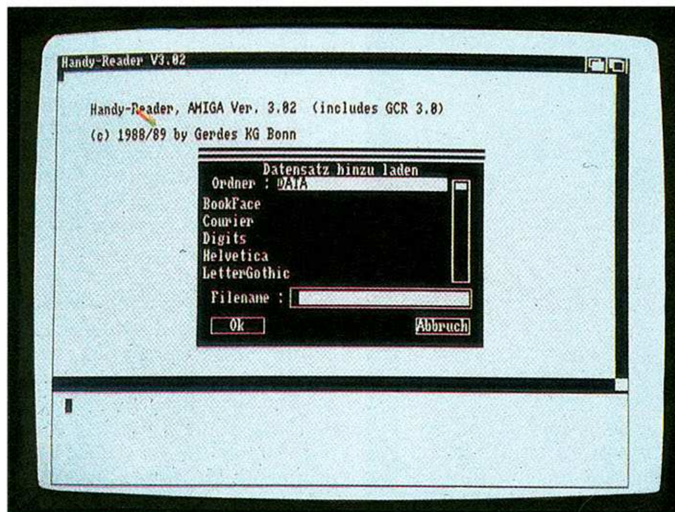


Bild 4. Der Bildschirm des 'Readers'. Hier können Texte 'gescannt' und in ASCII-Dateien umgewandelt werden

Beim Test zeigte sich, daß Schriften aus Magazinen teilweise zu klein und damit unleserlich sind, solange sie mit dem horizontalen Scan-Vorgang eingelesen werden. Erst bei Versuchen mit vertikalem Scannen stellten sich erste Erfolge ein. Das vertikale Scannen verlangt allerdings eine sehr viele höhere Präzision beim Halten des Scanners als das horizontale, weil der Scanner hier mit einer höheren Auflösung arbeitet.

An dieser Stelle ein paar Worte über die Handbücher: Beide, beim Farbscanner wie auch beim Schwarzweiß-Scanner sind in Deutsch und enthalten ausreichende Informationen über den jeweiligen Scanner und seine Anwendung sowie über die Programme. Gottseidank hat man auch nicht vergessen, für das Thema Texteinlesen ein konkretes Beispiel zu übernehmen. Anhand dieses Beispiels kann man nämlich den Umgang mit dem Scanner üben; auch wenn es beim ersten Mal nicht gleich klappt, der Erfolg ist mit dem Handbuch zumindestens gegeben.

Ob Bilder oder Textdateien – Scannen will gelernt sein

Scannen macht Spaß. Wer einmal mit dem Farbscanner Bilder aus Mutters Fotoalbum mit einem Grafikprogramm 'verhunzt' hat, will gar nicht mehr aufhören. Soviel zum Spaß an der Sache. Jetzt zum nötigen Ernst. Scanner sind eine teure Angelegenheit; der Schwarzweiß-Scanner mit seinem Preis von 798,– DM ist dabei noch wesentlich preiswerter als der Farbscanner (1498,– DM). Die Anwendungen der beiden sind ebenso unterschiedlich wie die Preise: Während der Farbscanner vor allem für Grafiker interessant sein dürfte oder für Anwendungen im Desktop-Publishing oder Desktop-Video, wird der Schwarzweiß-Scanner eher die interessieren, die Texte und Textvorlagen, die ausnahmsweise mal nicht auf dem Computer geschrieben wurden, auf eben diesem einsetzen wollen. Natürlich kann der Schwarzweiß-Scanner auch Grafiken scannen, wobei man dann natürlich auf Farbe verzichten muß.

Der Farbscanner ist ein 'Have fun'-Gerät, mit dem das Arbeiten ein (teures?) Vergnügen ist; der Schwarzweiß-Scanner ist ein nicht unproblematisches, aber dennoch nützliches, relativ neues Eingabegerät, mit dem man sich (preiswert?) viel Arbeit ersparen kann.

(jb)



Bild 5. Bei der 'Erkennen und Lernen'-Funktion wird der eingescannte Text anhand der Font-Dateien überprüft. Nicht erkannte Zeichen werden beim Anwender nachgefragt und ins Repertoire aufgenommen

AMIGA DOS Blitzlicht

Name: Handy-Scanner

Hersteller: Cameron

Vertrieb: Reisware

Preis:
Typ 10 (sw) 798,– DM,
Typ 6 (color) 1498,– DM

Positiv:
- leichte Installation
- gutes Handbuch
- Programme nicht kopiergeschützt (Installation auf Festplatte kein Problem)
- leichte Handhabung der Scanner
- gute Qualität bei Grafik (Colorscanner)
- verschiedene Scan-Modi beim Farbscanner
- Texterkennung des Schwarzweiß-Scanners gut

Negativ:
- 'Reader'-Programm bei Falscheingaben etwas eigenwillig (kein Schließen von Windows möglich)
- 'Scan & Save'-Programm etwas dürftig (speichert ohne Abfrage)

Digi View 4.0

Mit 4096 Farben im HiRes-Mode

Die Digi-View-Serie konnte bereits in der Vergangenheit mit beachtlichen Ergebnissen aufwarten. Mit Digi View 4.0 liegt nun ein Update dieser erfolgreichen Digitizer-Anwendung vor.

Digi View ist eine Soft- und Hardwarekombination, mit deren Hilfe Bildvorlagen und stehende Videobilder eingelesen und gespeichert werden können. Die Hardware besteht aus einem A/D-Wandler, der die Kamerasignale in für den Rechner verwertbare Informationen umsetzt, sowie einer Farbfilterscheibe, mit deren Hilfe die benötigten Rot-, Blau- und Grünauszüge erzeugt werden. Aufgabe der Software ist es nun, diese Auszüge wieder zu einem vollständigen Bild zusammenzupuzzlen.

Digi View unterstützt dabei so gut wie alle Auflösungen, falls genügend Speicher vorhanden ist. Bisher waren aber, zumindest in der HiRes-Auflösung, der Farbenpracht Grenzen gesetzt. Mit Hilfe des Software-Updates ist es nun möglich, auch im HiRes-Modus in 4096 Farben zu digitalisieren. Welche beeindruckenden Qualitäten damit zur Verfügung stehen, sehen Sie in Bild 2.

Auch wenn die geleisteten Änderungen nicht sofort ins Gesicht springen, die Verbesserungen gehen in die Tiefe.

Die Neuerungen im Detail

Da wäre zunächst der Filerequester, der dem von Digi Paint recht ähnlich sieht und eine wesentlich komfortablere Methode darstellt als sein Vorgänger. Außerdem wurden hier noch einige Funktionen wie Previous Dir, Laufwerkswahl und ähnliches implementiert.

Auch der Multitasking-Betrieb wurde ausgebaut. So ist es nun, genügend Speicher vorausgesetzt, möglich, vor dem Starten von Digi View Digi Paint zu laden und digitalisierte Images direkt in Digi Paint zu übernehmen. Auch an die Zukunft wurde gedacht, Digi View 4.0 ist mit einer Arexx-Schnittstelle ausgestattet, die die Fernbedienung des Programms und den Informationsaustausch mit anderen Programmen unterstützt.

Was das eigentliche Digitalisieren angeht, so können auch hier erfreuliche Verbesserungen verzeichnet werden. Zunächst wurde die Möglichkeit geschaffen, auch im HiRes-Modus mit Overscan in 4096 Farben zu arbeiten, sofern der Rechner über genügend Speicher verfügt.

Neben den bisher wählbaren Modi wie 16 C, 32 C, 4096 C, 4096+ C ist nun ein neuer Modus verfügbar, der von NewTek entwickelte Dynamic Mode. Dieser Modus soll noch schärfere Konturen und detailliertere Farbauflösung hervorrufen. Versuche haben gezeigt, daß NewTek diesem Anspruch auf jeden Fall gerecht geworden ist, wie Bild 2 recht anschaulich zeigt. Allerdings solange der Amiga rechnet, kann der unbedarfte Betrachter den Eindruck gewinnen, nur noch Grafiksalat zu sehen. Nur Geduld, sobald das Bild fertig berechnet ist, präsentiert es sich in seiner ganzen Farbenpracht und in hervorragender Qualität. Jedenfalls solange die Maus nicht bewegt wird, da dies recht bunte Effekte auf dem Screen hervorrufen kann. Der Verlust des Bildes muß bei Bewegung der Maus allerdings in Kauf genommen werden. In diesem Fall ist es ratsam, das Tastaturkommando für 'Bild speichern' im Kopf zu haben. Sehr angenehm gestaltet sich die Arbeit, wenn ein 68020-Prozessor vorhanden ist. Dieser kann die anfallenden Rechenzeiten erheblich verkürzen und trägt somit zum zügigen Arbeiten bei.

Als besonderes Bonbon enthält die Diskette mit dem 4.0 Update ein File namens Dynashow, mit dessen Hilfe digitalisierte Bilder zu einer Slideshow zusammengefaßt werden können. Wäre es nicht einmal ein netter Zug, den Freunden und Bekannten die letzten Urlaubsbilder als Slideshow zu verehren? Aber auch für Produktpräsentationen oder einfach kreative Arbeit ist die Digi-View-Serie wie geschaffen.

(mm)

AMIGA DOS Blitzlicht

Name: Digi View 4.0
Hersteller: NewTek Inc.
Vertrieb: Fachhandel
Preis: stand bei Redaktionsschluß noch nicht fest.

Positiv
- HiRes jetzt auch im 4096-Farben-Modus
- Darstellung im Dynamic Mode
- Unterstützung des 68020-Prozessor
- effektiver Filerequester
- Arexx-kompatibel
- Slideshow-Programm enthalten
- Datenaustausch mit Digi Paint im Multitasking-Betrieb
- ausführliches Handbuch

Negativ
- Maus darf im Dynamic Mode nicht bewegt werden

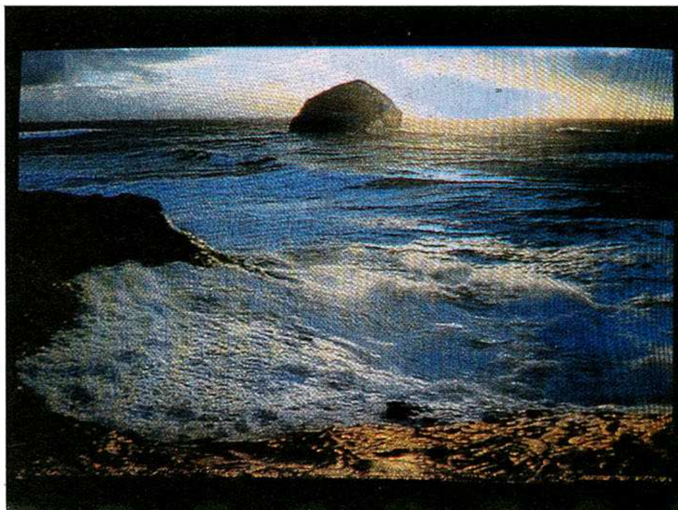


Bild 1. Digi-View-Gold im neuen Gewand. Neben der Unterstützung von 4096 Farben wurde auch ein vernünftiger Filerequester integriert

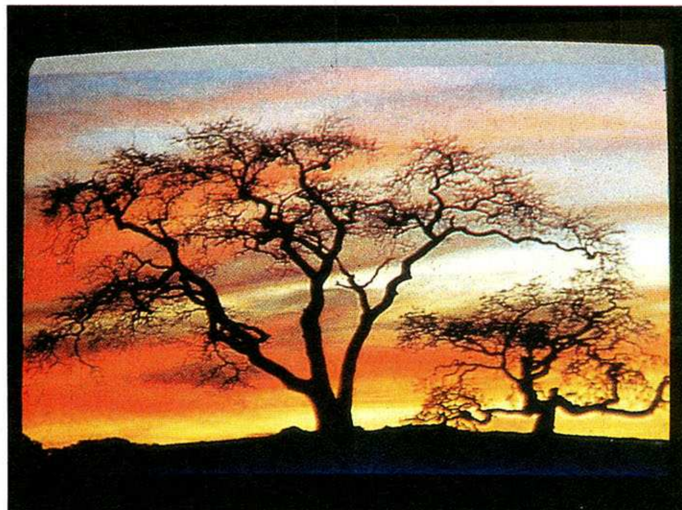


Bild 2. Die verbesserte Software kann der Hardware qualitativ hochwertige Bilder entlocken

Bei der Multifunktions-Tastatur Alphakey II von der Firma AFC können Sie Tastaturtreiber oder ähnliches getrost vergessen. Bestimmte Programmteile brauchen ab jetzt nicht mehr über komplexe Menüs oder diverse Tastenkombinationen aufgerufen zu werden, sondern können bequem durch einfachen Tastendruck angesprochen werden, wenn die Tastaturbelegung entsprechend angepaßt wurde.

Hierbei ist das Arbeiten mit den programmierbaren Einsteckmodulen von Vorteil. Diese Module dienen der Speicherung von Tastaturbelegungen, wobei zwischen drei Arten von Modulen unterschieden wird.

Die **Memory-Module** haben freien Speicherplatz, der zum Abspeichern von Tastaturbelegungen zur Verfügung steht. Alle gestaltbaren Tasten aus allen Ebenen (dazu später mehr) können hier abgespeichert werden. Diese Memory-Module können jederzeit gelöscht und neu beschrieben werden; der Inhalt bleibt auch nach dem Entfernen des Moduls oder nach dem Ausschalten des Rechners erhalten.

Fix-Module enthalten eine ab Werk vorprogrammierte Tastenbelegung. Diese Belegung ist nicht veränderbar und auf spezielle Software abgestimmt.

Eine Kombination aus Memory- und Fix-Modul stellen die **Kombi-Module** dar. Sie enthalten sowohl eine vorprogrammierte Tastenbelegung als auch Speicherplatz, um selbst Tasten zu definieren.

Unbefugte müssen draußen bleiben – der Passwortschutz

Um eine vorprogrammierte Tastaturbelegung zu laden, genügt es, das Modul in den Schacht zu stecken. Daraufhin wird der Inhalt des Moduls in den Arbeitsspeicher der Tastatur geladen. Nach dem Ladevorgang kann das Modul dann wieder aus der Tastatur genommen werden. Natürlich können auch Module kopiert werden. Dazu lädt man den Inhalt einer Cartridge in den Tastatur-Arbeitsspeicher, entfernt das Master-Modul und überträgt den Inhalt des Speichers auf ein neu eingestecktes Modul.



Die Multifunktions-Tastatur Alphakey II von AFC Computer – ein professionelles Keyboard mit programmierbaren Steckmodulen

Alphakey II

Eine Tastatur für Profis

In der Ausgabe 2'90 haben wir im Rahmen der Aktuell-Seiten eine Alternative zur Standard-Amiga-Tastatur vorgestellt. Das auffallendste an diesem Keyboard sind seine Steckmodule, die eine individuelle Anpassung an jedes beliebige Programm ermöglichen.

Zwei Funktionen der Alphakey II können über einen Passwortschutz gesperrt werden: zum einen die gesamte Ausgabe an den Rechner und zum anderen die Gestaltungsfunktion der Tastatur. Die dafür notwendigen Passwörter werden in den Modulen abgespeichert, so daß benutzer- und programmabhängige Sicherheitsstufen aufgebaut werden können. Die vergebenen Passwörter sind modulabhängig. Sobald eins der mit einem Passwort versehenen Module in die Tastatur gesteckt wird, wird die jeweilige Funktion sofort gesperrt und ist erst nach Eingabe des Wortes oder nach Herausziehen des Moduls wieder zugänglich.

Ideal für das Arbeiten unter MS-DOS

Besonders für Amiga-Benutzer, die ihren Computer mit einem Sidecar (Amiga 1000) oder der XT- beziehungsweise AT-Karte auf dem Amiga 2000 betreiben, werden die Funktionsvielfalt der Alphakey II zu schätzen wissen. Die Alphakey II bietet drei Funktionsebenen, die mit bestimmten Tasten aktiviert werden können. Verwendet man ein Fix- oder ein Kombi-Modul, steht eine vierte, fest vorprogrammierte Ebene zur Verfügung. Dabei zeigen Leuchtdioden an den Funktionsebenen-Tasten die jeweils aktive Ebene an.

Die Ebenen zwei und drei (die erste entspricht der normalen Rechnertastatur, im folgenden kurz PC-Ebene genannt) können frei belegt werden. Belegen Sie einfach die zweite Ebene für das Arbeiten mit Wordstar, die dritte Ebene beispielsweise für den Umgang mit Multiplan, so können Sie wahlweise per einfachem Tastendruck zwischen diesen drei Ebenen umschalten. Ein kleines Beispiel soll dies verdeutlichen:

Sie arbeiten in Wordstar und befinden sich in der zweiten Ebene, möchten aber während der Bearbeitung einer Textdatei zum Korrekturlesen einer bestimmten Textstelle eine Hardcopy von einer Bildschirmseite auf dem Drucker erstellen. Die dazu notwendige Taste PrtSc liegt jedoch in der PC-Ebene und ist nicht erreichbar. Kein Problem für Alphakey II, mit einer kurzen Tastenkombination aktivieren Sie die PrtSc-Taste der PC-Ebene, ohne die Wordstar-Ebene zu verlassen.

Die Alphakey II bietet drei integrierte Systeme für den Anschluß von Datenerfassungsgeräten an die Tastatur und deren Einbindung in die vorhandene Software. Sie ist mit KCC- und ADR-Systemen ausgerüstet. KCC bedeutet Key Code Conversion (Tastencode-Umwandlung), während ADR für ein Schnittstellensystem steht, das die Einstellung von Baudraten und anderen Übertragungsformaten überflüssig macht, da die angeschlossenen ADR-fähigen Datenerfassungsgeräte sofort erkannt und alle Parameter automatisch abgestimmt werden. Dies ermöglicht dem Anwender beispielsweise, einen Barcodeleser anzuschließen, dessen Daten an die Tastatur übermittelt werden, die diese dann umwandelt und dann an den Rechner schickt.

Das Testergebnis: Gut – aber teuer

Ergonomisch gesehen, ist die Tastatur durch eine Handballenauflage vorteilhaft, langes Arbeiten am Rechner strengt somit weniger an. Das hat aber auch den Nachteil, daß die Abmaße recht groß sind und die Tastatur somit viel Arbeitsfläche in Anspruch nimmt. Der Benutzer kann zwar ein Hilfsmenü aufrufen, in dem Dinge wie Baudrate, Repeat-Verzögerung, Aktivieren beziehungsweise Deaktivieren des Keyklicks (bei Tastendruck ertönt ein Ton) zu erfahren ist, jedoch fehlt mir eine Anzeige der vorprogrammierten Tasten. Auch scheint der Preis von knapp 1500 DM doch etwas überzogen.

(br)

AMIGA DOS Blitzlicht

Name: Alphakey Amiga

Hersteller: AFC hard and software GmbH
Theodor Heuss-Ring 52
5000 Köln 1

Preis: zirka 1500 DM

Positiv:
- Tastaturbelegungen in Modulen abspeicherbar
- guter Druckpunkt
- mehrere Funktionsebenen
- Passwortschutz

Negativ:
- Tasten ein wenig klein
- hoher Preis
- Tastatur benötigt viel Platz

Schieben statt rollen

Erfahrungen mit der 'Lichtmaus'

Kurz nach der Messe kam sie in die Redaktion – die erste kugellose Maus für den Amiga. Was ist dran an so einem optischen Eingabegerät? Wir wollten es genauer wissen und begannen sofort mit der 'Schieberei'.

Das erste, was einem bei der neuen Maus auffällt ist, daß sie ihr Mousepad, also ihren Untergrund gleich mitbringt. Das Mousepad hat aber eine Doppelfunktion, die einem technisch nicht so versierten Menschen nicht gleich ins Auge springt: Es teilt sich in eine sehr kleines schwarzweißes Raster auf, welches die Maus für ihre Arbeit benötigt. Ohne das Pad ist an eine Bewegung des Mauszeigers gar nicht zu denken. Damit dürfte wohl auch klar sein, daß unsachgemäße Behandlung des Pads der ganzen Maus schadet. Anders gesagt: Kaffee-, Kakao- oder Cola-Flecken auf dem Pad, und schon ist es aus mit der Arbeit.

Sie ist leichter und ergonomischer – die Optomaus

Die optische Maus liegt leichter in der Hand als ihre Kollegen, wobei auch klar ist, warum. Die fehlende Kugel macht's, daß die Bewegung des Mauszeigers nicht mehr zur Arbeit wird. Die Maustasten enthalten Mikroschalter und sind ebenfalls gut zu bedienen. Die Führung auf dem Pad ist präzise, fehlerhafte Abweichungen waren nicht festzustellen. Der Anschluß ist genauso wie bei der Original-Maus; einfach den Stecker in den Port, und schon kann es losgehen.

Während bei der mechanischen Maus die Kugel eine Mechanik antreibt, die wiederum zwei Lichtschranken zur Impulsabgabe bringen, ist das Prinzip der Optomaus völlig anders:

Eine Leuchtdiode gibt einen Lichtstrahl (meistens Infrarot-Licht auf eine reflektierende Fläche ab, die dieses Licht auf einen Empfänger zurückwirft.

Der Unterschied zwischen optischer und mechanischer Maus

Wenn der Reflektor, in unserem Fall das Pad, aus sich abwechselnden schwarzen und weißen Flächen besteht, so wird das Licht unterschiedlich stark reflektiert. Die Folge: Statt einer einheitlichen Lichtstärke entsteht ein flackernder Lichtstrahl, der beim Empfänger, der wiederum den einkommenden Lichtstrahl in Stromimpulse umwandelt, eine Impulsfolge erzeugt. Um nun die Bewegungsrichtung festzustellen, werden zwei Empfänger im Winkel von 90 Grad zueinander gesetzt. Die Impulse, die auf beiden Empfängern eintreffen, werden folgerichtig zugeordnet, so daß eine Bewegungsrichtung festgelegt werden kann.

Daß die Original-Maus ihre Nachteile hat, wissen viele Benutzer aus leidvoller eigener Erfahrung. Vor allem die Tasten sind ein Schwachpunkt. Aber ist der Kauf einer neuen, andersartigen Maus gerechtfertigt?

Die Optomaus von Konyo liegt gut in der Hand, ist leicht und ohne Probleme anschließbar.

Ein Vergleich zwischen beiden Mäusen

Dies sind die Punkte, die man als positiv bewerten kann. Bei längerem Gebrauch kann es Probleme mit dem Pad geben, wenn erste 'Verschleißerscheinungen' in Form von Flecken entstehen. Die Maus selbst ist ein guter Ersatz für die Original-Maus. Sie besitzt drei Tasten, von denen jedoch nur die beiden äußeren benutzbar sind. Ist das Pad sauber, fährt sie leicht und ohne Problem auf ihm. Sie ist ziemlich verschleißfrei, da die mechanische Übersetzung der Kugeldrehung in getrennte Drehbewegungen für horizontale und vertikale Richtung fehlt.

Diese zumindestens im mechanischen Bereich bestehende Verschleißfreiheit und die Tatsache, daß die Tasten mit Mikroschaltern ausgerüstet wurden, machen die Maus zur guten Alternative.

Leider stand der Endverbraucherpreis bei Redaktionsschluß noch nicht fest, so daß hier kein Kosten-Nutzenfaktor erkennbar ist, trotzdem kann die Optomaus ohne Furcht in den Kampf um den Stammplatz am Amiga gehen.

(jb)

AMIGA DOS Blitzlicht

Name: Optische Maus
Anbieter: Konyo
Preis: stand bei Redaktionsschluß noch nicht fest

Positiv:
- leicht
- liegt gut in der Hand
- Mikroschalter für die Tasten

Negativ:
- nur mit dem Pad funktionsfähig
- Pad nicht stabil genug, knickt leicht



Bild 1. Während bei herkömmlichen Mäusen eine Kugel über eine Mechanik zwei Lichtschranken steuert, die wiederum ihre Impulse über eine Elektronik an den Rechner weitergibt...

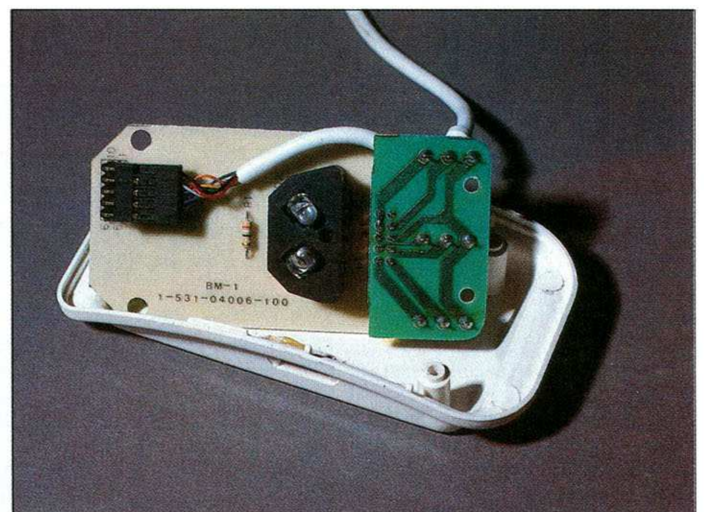


Bild 2. ...befindet sich in der optischen Maus eine Lichtsteuerung, welche die benötigten Impulse über ein Raster (was sich sinnigerweise auf dem Mousepad befindet) erzeugt und so an den Amiga weitergibt

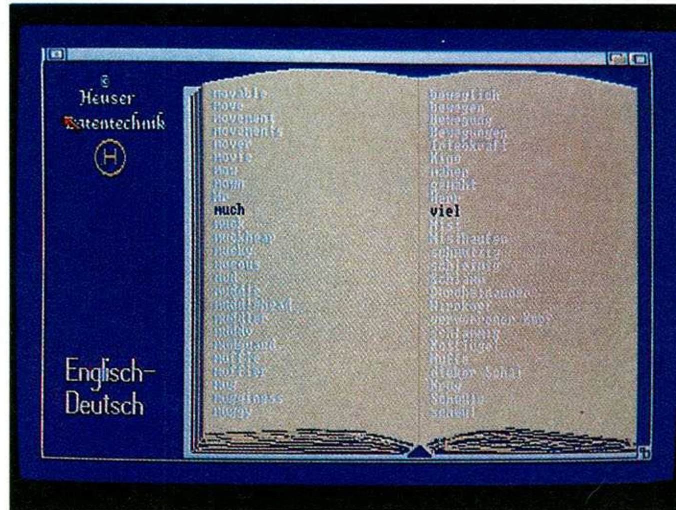


Bild 1. Amiga, übersetz' doch mal

Andreas Polk

Das digitale Wörterbuch

Lernen leicht gemacht?

Fremdsprachen auf herkömmliche Weise zu erlernen, ist oft trocken und öde. Der Weg zur Sprache führt nun weg vom Lehrbuch und hin zum Computer.

Die Firma Heuser Datentechnik wirbt mit recht kleinen Anzeigen für ein digitales Wörterbuch, mit dem Vokabeln in sekundenschnelle nachgeschlagen werden sollen. Ob das Programm dies erfüllen kann, soll ein Test entscheiden. Das Wörterbuch kann man in zwei Fassungen erwerben, Englisch-Deutsch und Deutsch-Englisch. Zu beiden Disketten erhält man ein dünnes Handbuch, das etwas zu knapp gehalten ist.

Praktischer Nutzen

Bootet man die Diskette auf einem 512-kByte-Rechner, so wird das Wörterbuch direkt gestartet. Ansonsten muß erst auf der Workbench das entsprechende Icon angeklickt werden. Zu loben ist, daß das Programm auf allen Rechnern läuft. Da die Wörterbuchdatei ganz in den Speicher geladen wird, stößt man bei 512 kByte schon schnell an die Grenzen. Ergänzungen zum Wörterbuch sind lediglich begrenzt möglich.

Bereits während der Eingabe sucht das Wörterbuch aus den vorhandenen Vokabeln nach Übereinstimmungen. So nähert man sich schrittweise dem gesuchten Begriff. Kann ein Wort nicht gefunden werden, so wird dasjenige angezeigt, was laut alphabetischer Reihenfolge davorsteht. Hat man die Vokabel gefunden, so kann man sie auch ausdrucken lassen. Möchte man das Wörterbuch um eine Vokabel erweitern, so muß dazu ein entsprechendes Programm aufgerufen werden. Man gibt die neue Vokabel und deren Bedeutung an und startet dann den Einfügevorgang. Auch Umlaute werden akzeptiert.

Mit dem Programm 'Transfer' kann man Listen von Wörtern übersetzen lassen. Das ist dann sinnvoll, wenn man nicht alle Vokabeln einzeln nachschlagen möchte. Man muß lediglich alle Vokabeln in eine Datei schreiben und diese Datei übersetzen lassen. Jedoch sind hier auch Grenzen gesetzt, denn der Versuch, einen englischen Text übersetzen zu lassen, scheiterte kläglich. Dabei kam nur Wortsalat heraus. Dies ist auch sicherlich nicht das Anwendungsgebiet des Programms.

Wer Vokabeln trainieren möchte, kann dies auch mit dem eingebauten Trainer tun. Hier werden allerdings die Vokabeln durcheinander abgefragt, eine Abstufung nach Schwierigkeitsgrad wäre sicherlich sinnvoll. Positiv ist anzumerken, daß auch einzelne Vokabeln ausgedruckt werden können.

Vom Konzept her ist das Programm wirklich nicht schlecht gemacht. Allerdings unterliegt man immer der Einschränkung von 20000 Vokabeln, denn wer gibt schon noch weitere 20000 Vokabeln ein?

Wermutstropfen

Daß dies nur ein minimaler Auszug des gesamten englischen Wortschatzes ist, dürfte jedem klar sein. So wird für Wörter, die mehrere Bedeutungen haben, oft nur eine angegeben. Auch fehlen nicht so gängige Vokabeln in der Datei. Hier muß man weiterhin auf das Buch zurückgreifen. Diesem Nachteil könnte man aus dem Weg gehen, indem man die Vokabeldatei auf der Festplatte ablegt. Das Suchen dauert dann zwar länger, die Vokabeldatei könnte aber wesentlich umfangreicher werden.

Übrig...

...bleibt somit ein gemischtes Gefühl. Auf der einen Seite muß man den hohen Preis (rund 130,- DM für beide Bücher, ein Langenscheidt kostet beispielsweise unter 50,-DM) und den begrenzten Wortschatz bedenken, auf der anderen Seite steht aber das schnelle Finden von Vokabeln. Möchte man kurz etwas nachschlagen, wie zum Beispiel beim Lesen eines Handbuchs, so erweist sich dieses Programm als große Hilfe. Vokabeln sind wesentlich schneller aufzufinden als im Wörterbuch und der Sinn ist meistens auch klar. Will man allerdings einen Text genau verstehen und alle Übersetzungsmöglichkeiten eines Wortes kennen, so ist es ratsamer, zum guten alten Buch zu greifen, denn hier ist das digitale Wörterbuch doch etwas überfordert.

(vb)

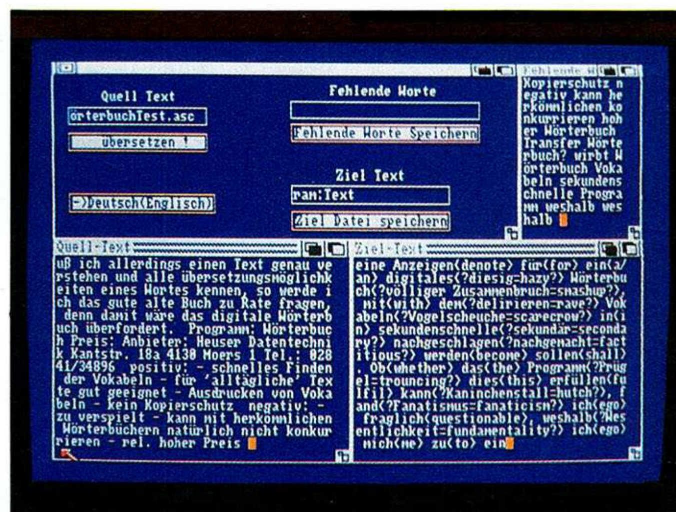


Bild 2. Mit Transfer werden ganze Dateien übersetzt

AMIGA DOS Blitzlicht

Name: Wörterbuch
Preis:
Englisch/Deutsch 69,– DM
Deutsch/Englisch 59,– DM
Anbieter:
Heuser Datentechnik
Kantstr. 18a
4130 Moers 1
Tel.: 02841/34896

Positiv:
- schnelles Finden der Vokabeln
- für 'alltägliche' Texte gut geeignet
- Ausdrucken von Vokabeln
- kein Kopierschutz

Negativ:
- zu verspielt
- kann mit herkömmlichen Wörterbüchern natürlich nicht konkurrieren
- relativ hoher Preis

AMIGA-DOS-Pixel-Panorama

Die Macht der bunten Bilder ist ungebrochen. Bilder, ob bewegt oder unbewegt, können wie kein anderes Medium Eindrücken und Phantasien Gestalt verleihen. Stellt jedes gemalte Bild nicht auch ein Abbild der Phantasien des Künstlers dar?

Mitnichten, was ist mit Kopien, Fälschungen und Plagiaten? Sind hier nur gute Handwerker bei der Arbeit, zwar technisch versiert, aber ohne die Imagination des Künstlers; Menschen also, die vor der Aufgabe, eigene Bilder zu kreieren, kapitulieren müssen? Hier stellt sich automatisch die Frage: Was ist Kunst?

Eine Frage, deren Antwort auf so unsicheren Fundamenten wie Geschmack und Interpretation basieren muß. Und wie sieht es mit der Gebrauchskunst aus, mit der wir allenthalben konfrontiert werden? 'Nein Danke' von Heinz Eger kann ein gutes Beispiel für Gebrauchskunst sein. Tatsächlich könnte das Bild unverändert als Nichtraucherplakat örtlicher Krankenkassen Verwendung finden.

Rainer Appels 'Robopunk' und 'Spinne' sind ausgezeichnete Beispiele für kreative Raytracing-Arbeiten. Zwar kann so gut wie jedes Bild, das durch Raytracing erzeugt wird, auch mittels eines ordinären Grafikprogramms erstellt werden; das Flair und die Farbgebung eines Raytracing-Bildes würde jedoch nicht erzielt. Das letzte Bild stammt von M. Billerbeck, der sich einen Nissan als Vorlage erkoren hat.

Das Pixel-Panorama steht natürlich auch weiterhin für alle Amiga-Künstler offen. Wie bisher gibt es für die Veröffentlichung Ihrer Bilder ein Dankeschön in Form eines Softwarepreises im Wert von zirka 100 DM. Mitmachen ist ganz einfach:

Packen Sie Ihre Bilder auf eine Diskette, und vermerken Sie auf dem Etikett der Diskette neben Ihrer kompletten Anschrift auch das Grafikformat der Bilder bzw. die Grafikprogramme, die bei der Erstellung verwendet wurden. Das Ganze stecken Sie dann in einen Umschlag und schicken es an den:

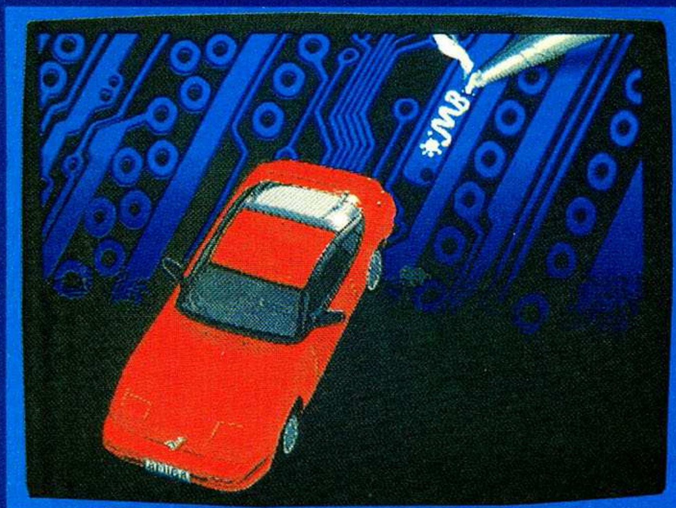
DMV-Verlag
Redaktion AMIGA DOS
Kennwort Pixel
Postfach 250
3440 Eschwege

Vielleicht hatten Sie beim Betrachten der Bilder auf dieser Seite selbst eine zündende Idee, die ein feines Bild werden könnte. Aber vergessen Sie nicht: Wenn Sie selbst Ihre Ideen nicht umsetzen, tut das wohl auch niemand anders. In diesem Sinne viel Spaß an Ihrem Amiga!

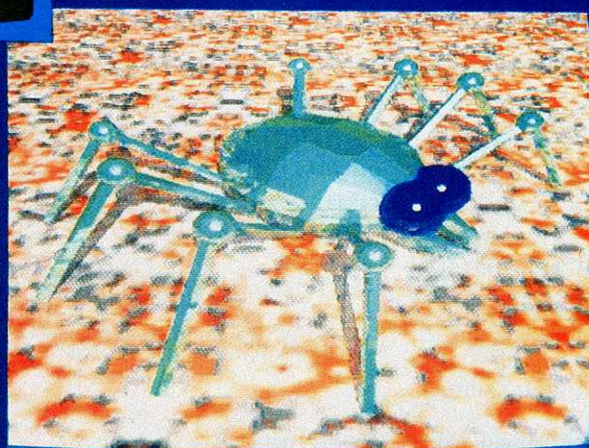
(hs)



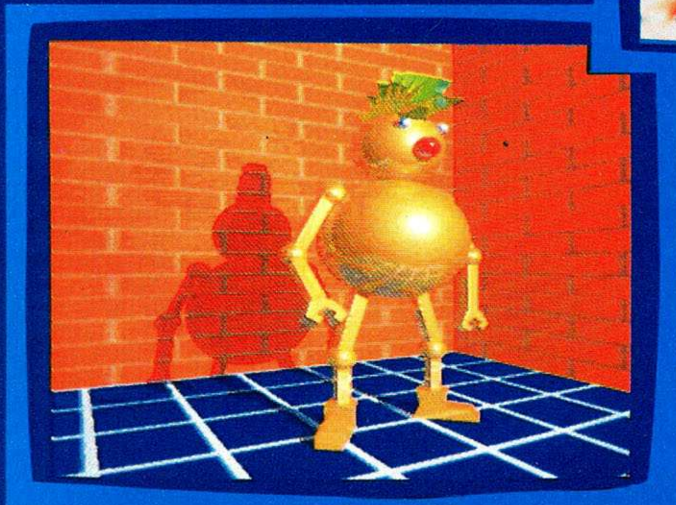
Aufruf zum Nichtrauchen



Nissan von M. Billerbeck



Spinne, wie auch Robopunk eine Raytracing-Kreation



Robopunk



Butcher – Arbeiten mit dem Bilderbieger

Jeder, der sich intensiver mit Grafik auf dem Amiga beschäftigt, wird über kurz oder lang neben dem künstlerischen Aspekt der Materie auch den technischen kennenlernen. Dabei stellt sich heraus, daß ein Computer ein weit komplizierteres Medium ist als ein Blatt Papier oder eine Leinwand.

Kursfahrplan

Teil 1: Grafikgrundlagen – Amiga und Butcher

Teil 2: Butcher – Bedienung ohne Kompromisse

Nicht nur Werkzeuge zur bequemen Erstellung von Bildern sind gefragt, sondern auch Programme, die dem Anwender die Kontrolle über die technische Seite der Computergrafik verleihen.

Eines der am weitesten verbreiteten Hilfsprogramme zur Bildbearbeitung ist der Butcher von Eagle Tree Software. Das Programm, obwohl zur Grafikbearbeitung ausgelegt, ist alles andere als ein 'Malprogramm'. Zwar sind auch rudimentäre Zeichenwerkzeuge enthalten, jedoch dient Butcher vielmehr der Manipulation der Bilddaten. Insbesondere kann der Anwender mit ihm die Bitplanes und Farbregister des Amiga beeinflussen.

Nähere Informationen über den technischen Aufbau von Computerbildern auf dem Amiga sowie Bitplanes und Farbregister finden Sie im ersten Teil dieser Werkstattserie in Ausgabe 6 '90.

Das <Effect>-Menü

In diesem Menü befinden sich im wesentlichen Werkzeuge, deren Wirkung sich bei den Farben eines Bildes bemerkbar macht. Im Grunde ließe sich jede dieser Funktionen auch über das <Make Palette>-Menü realisieren, würde auf diese Weise jedoch viel mehr Arbeit erfordern.

<Toner>
Im fotografischen Bereich erfreuen sich Trickfilter großer Beliebtheit. Diese Trickfilter bestehen aus einer gefärbten Linse. Wird eine Aufnahme durch eine solche farbige Linse gemacht, z.B. eine rotgefärbte, entsteht ein Bild, das sich ausschließlich aus Rottönen zusammensetzt. Wird die <Toner>-Funktion auf ein Bild angewendet, resultiert daraus ein Farbeffekt, der das behandelte Bild so aussehen läßt, als sei es durch einen Trickfilter dargestellt. Mittels der beiden Schieberegler können Farbwert und Farbintensität (Hue und Saturation) manipuliert werden. Betrachten Sie nun die Resultate von <Toner> im <Make Palette>-Requester, werden Sie feststellen, daß alle Register mit unterschiedlich hellen Abstufungen eines Farbtons gefüllt sind. Unmittelbar neben den Schiebereglern befinden sich drei Funktionsfelder, <Restore>, <Keep> und <Cancel>.

<Restore> macht alle seit dem Aufrufen des <Toner>-Requesters getätigten Änderungen ungeschehen.

<Keep> schließt den <Toner>-Requester, wobei alle getätigten Änderungen als aktuelle Einstellung übernommen werden.

<Cancel> verläßt den <Toner>-Requester, ohne die getätigten Änderungen zu übernehmen.

<Pos < - > Neg>
Dieser Menüpunkt bewirkt, daß die Palette ins negative verkehrt wird. Ein nochmaliges Benutzen dieser Funktion stellt den Urzustand wieder her.

<Complement>
Die Funktion <Complement> ähnelt der <Pos < - > Neg>. Nur werden hier die Farben nicht ins negative verkehrt, sondern in ihr Komplementär-Äquivalent umgewandelt. Wird beispielsweise die Farbe Grün mit <Pos < - > Neg> behandelt, erscheint an Stelle des Grüns ein heller Rosaton. Wird dasselbe Grün mit <Complement> behandelt, ist das Resultat ein volles farbgesättigtes Rosa.

<False Colours>
Auch diese Funktion wirkt auf alle Farbregister und bewirkt eine komplette Verschiebung der Farbwerte. Um bei unserer Beispielfarbe Grün zu bleiben: Bei der ersten Nutzung von <False Colours> wird aus Grün Blau. Bei der nächsten Aktivierung wird aus Blau Rot, aus dem bei einer weiteren Benutzung von <False Colours> wieder Grün wird.

<Pseudo Colours>
wandelt die aktuelle Palette so um, daß primär Grün- und Rottöne enthalten sind.

<Black & White>
wandelt die aktuelle Palette in eine Schwarzweißpalette um. Zwischentöne werden als Graustufen dargestellt. Die Resultate ähneln einer Schwarzweißfotografie.

<Antique>
wandelt alle Farben des aktuellen Bildes um. Es entsteht der Effekt einer antiken Fotografie.

<Adjust>, <Separate>, <Band>

Die nächsten drei Punkte dieses Menüs <Adjust Separation>, <Separate> und <Band> sind durch ihre Funktion miteinander verbunden und dienen dazu, Farbausdrucke zu optimieren. Bilder, die solcherart ausgedruckt werden sollen, werden durch den Druckertreiber in einzelne Farbmasken zerlegt (natürlich gilt dies nicht für alle Drucker, ist jedoch die gebräuchlichste Methode). Jede dieser Farbmasken entspricht nun einem kompletten Bild, in dem alle Bildpunkte in einer Farbe gesetzt sind.

Ein Beispiel: Um die Farbe Orange zu erzeugen, muß ein Farbdrucker zunächst eine gelbe Fläche drucken, über die dann deckungsgleich eine rote gedruckt wird. Die beiden übereinanderliegenden Farben erscheinen dem Betrachter nun wie Orange.

Durch den Menüpunkt <Separate> gelangt der Anwender in ein Untermenü, in dem er die jeweilige Grundfarbe, von der eine Farbmaske angefertigt werden soll, auswählen kann. Um möglichst große Vielfalt zu gewähren, stehen sieben verschiedene Grundfarben zu Auswahl: <Yellow>, <Black>, <Magenta>, <Cyan>, <Red>, <Green>, <Blue>.

Neben einer einfachen Vorkontrolle der Farbmasken, die dem Drucker übergeben werden sollen, können die Funktionen <Separate> und <Adjust Separation> auch anderweitig benutzt werden. Wer sich die sieben unter <Adjust Separation> zur Verfügung stehenden Farben einmal genauer ansieht, wird feststellen, daß hier einerseits die Grundfarben jedes Bildes vorhanden sind, die bei einer Darstellung auf einem Monitor nötig sind (Rot Grün, Blau) und andererseits die vier Grundfarben, die bei einer drucktechnischen Darstellung benötigt werden (Gelb, Magentarot, Cyanblau und Schwarz). Zusammen mit entsprechender Hardware können mit diesen Funktionen sogar Offsetdruckvorlagen für Vierfarbdrucke erstellt werden.

Der letzte Menüpunkt, <Band>, dient der Überprüfung der Intensität der einzelnen Farbmasken.

Das <Process>-Menü

Hier findet der Anwender Mittel und Wege, Grafiken auf die phantastischsten Arten zu manipulieren. Es gibt Tools, um ein Bild so zu manipulieren, daß der Eindruck eines Mosaiks entsteht. Bilder können in ihre Bitplanes zerlegt werden und allerlei andere Dinge mehr. Aber sehen Sie selbst.

<Edges>

Nach dem Anwählen des ersten Menüpunktes öffnet sich auf der Zeichenfläche ein besonderes Window, das wiederum eine Reihe von Funktionen und Reglern enthält. Die linke Seite des Windows wird durch zwei Schieberegler, das <Paper>-Gadget und eine Darstellung der aktuellen Palette dominiert. Die beiden Regler, mit <Threshold> und <Color Variance> bezeichnet, dienen der Feinabstimmung zweier wichtiger Parameter. <Threshold> und <Color Variation> bezeichnen die Faktoren für Reaktionsschwelle und Farbintensität nach der eine später aus dem <Edge>-Window aktivierte Funktion arbeiten muß.

Im <Paper>-Gadget kann der Benutzer die Farbe der Zeichenfläche markieren. Die in diesem Register enthaltene Farbe wird sozusagen als Maloberfläche gewertet.

Auf der rechten Seite des Windows finden sich <Map>, <Enhance>, <-0+>, <Color>, <View>, <Clip>, <Restore>, <Keep> und <Cancel>.

Um die Möglichkeiten von <Edge> zu erklären, muß ein bißchen weiter ausgeholt werden. Grundsätzlich dient <Edge> dazu, einzelne Farben bzw. Farbregister in einem Bild zu manipulieren, und zwar in einer Art und Weise, die die Möglichkeiten von <Make Palette> weit übersteigt.

<Map> sucht eine vorher in der Palettenanzeige definierte Farbe aus dem aktuellen Bild und betont sie, indem umgebende Pixel in die definierte Farbe konvertiert werden. Das Resultat sieht wie ein Steinrelief aus. Je weiter nun die beiden Regler <Threshold> und <Color Variance> aus der Nullstellung herausbewegt werden, desto gröber wird der erzielte Reliefeffekt.

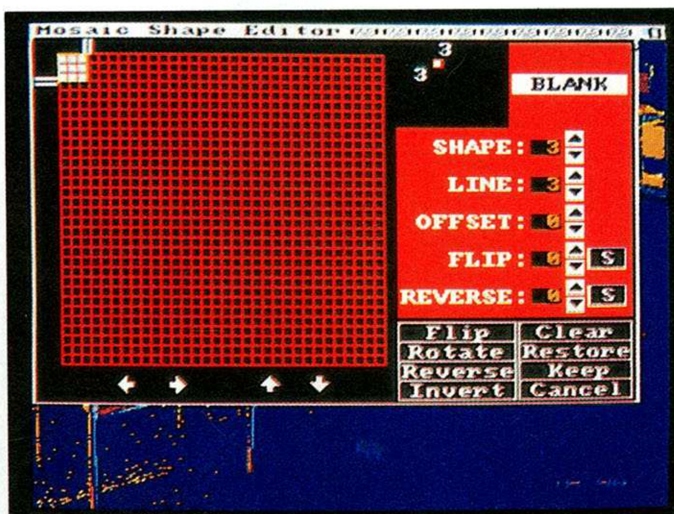
Wurde ein Bild mittels <Map> quasi in eine einfarbige Reliefstudie verwandelt, verschwindet bei diesem Prozeß auch das ursprüngliche Bild. Mit <Enhance> kann nun das ursprüngliche Bild zu der Reliefdarstellung geblendet werden. Dabei 'übermalt' der Butcher nur die als Papierfarbe definierten Pixel. Das 'Reliefbild' erscheint so, als würde es über das eigentliche Bild gelegt werden.

Der mittels <Map> erzielte Effekt kann durch die drei Schalter <->, <0> und <+> beeinflußt werden.

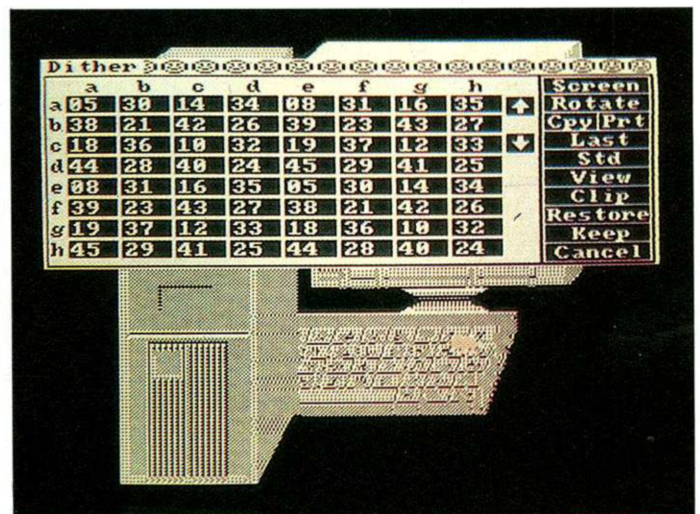
<-> bewirkt, daß die Reliefstruktur von rechts oben nach links unten verläuft, <+> läßt die Reliefstruktur von links oben nach rechts unten führen, während <0> eine senkrechte Ausrichtung bewirkt. Weitere Effekte lassen sich erzielen, wenn die Funktion mit verschiedenen <Threshold>- und <Color Variance>-Reglerstellungen ausprobiert wird.

Mit <View> kann der Anwender das <Edge>-Window ausblenden, um die aktuelle Grafik ungestört begutachten zu können. Ein Druck auf die rechte Maustaste bringt das <Edge>-Window wieder auf den Monitor. <Clip> definiert einen beliebigen Bildausschnit der aktuellen Grafik. Dieser Ausschnitt kann nach Belieben aus- und eingeschaltet werden. Ist der Bildausschnitt aktiv, wirkt die <Map>- und <Enhance>-Funktion nur noch auf der Fläche des definierten Auschnittes.

<Restore> beseitigt bereits getätigte Änderungen und stellt die zu bearbeitende Grafik in der Form wieder her, in der sie sich vor dem Aufruf



Hier können eigene Mosaikmuster definiert werden, der <Mosaik Shape Editor>



Farbige Vorlagen in Raster zerlegen, die <Dither>-Funktion macht es möglich

des <Edge>-Windows befand.

<Keep> verläßt das <Edge>-Window, wobei alle getätigten Änderungen ins aktuelle Bild übernommen werden.

<Cancel> verläßt das <Edge>-Window, wobei alle hier getätigten Einstellungen verworfen werden.

<Filter>

Auch der zweite Punkt des <Process>-Menüs, <Filter>, öffnet ein Window mit verschiedenen Funktionen. Mit den Funktionen des <Filter>-Windows ist es dem Anwender möglich, Pixel einer bestimmten Farbe 'auszufiltern'. In dem Window findet der Anwender eine komplette Palettendarstellung und die Optionen <Filter>, <Pick>, <Clip>, <Restore>, <Keep> und <Cancel>.

Unmittelbar über der Palette, befindet sich ein Gadget, in dem der Anwender den 'Filterfaktor' einstellen kann. Je höher der hier eingestellte Wert, desto mehr Pixel der betreffenden Farbe werden ausgefiltert.

Während des Filterprozesses werden die Pixel, die in der zu filternden Farbe gezeichnet sind, durch die angrenzende Farbe ersetzt.

Mit <Pick> kann sich der Benutzer alle Bildpunkte, die in einer bestimmten Farbe gehalten sind, anzeigen lassen. Die betreffende Farbe beginnt zu blinken.

Wie im <Edge>-Window auch, kann bei aktivem <Filter>-Window ein Bildausschnitt <Clip>, definiert werden. Ist ein solcher Bildausschnitt aktiv, wirken alle getätigten Aktionen nur auf den definierten Bildausschnitt.

<Restore>, <Keep> und <Cancel> haben dieselbe Funktion wie im <Edge>-Window.

<Mosaik>

Diese Funktion wandelt das aktuelle Bild in ein Mosaikbild um. Die Wirkungsweise der <Mosaik>-Funktion läßt sich mittels <Shape Mosaik> definieren.

<Shape Mosaik>

Hinter diesem Menüpunkt verbirgt sich ein Untermenü, in dem der Anwender auf die Optionen <Load>, <Save> und <Edit> stößt. Während <Load> und <Save> Windows öffnen, über die bereits erstellte Mosaikformen abgespeichert und wieder eingelesen werden können, steckt hinter <Edit> ein spezieller Editor, mit dem eigene 'Mosaik-Grundformen' definiert werden können.

Die Möglichkeiten der Bildbearbeitung durch Mosaik sind reichhaltig. Elementar betrachtet, erfolgt nach dem Aktivieren der Funktion eine Zerlegung des aktuellen Bildes in kleinere Teilstücke, gerade so, als würde die horizontale und vertikale Auflösung des Bildes reduziert. Jedoch wird nach wie vor dasselbe Bild gezeigt. Butcher ermittelt aus jedem Teilfeld der neu entstandenen Matrix den dominierenden Farbton (der Farbton, der in dem betreffenden Teilstück am meisten enthalten ist) und füllt damit das komplette Teilstück des Bildes.

Mit der Option <Edit> im <Mosaik>-Untermenü wird der <Mosaik Shape Editor> aufgerufen. Der <Mosaik Shape Editor> besteht aus einer 32 x 32 Punkte Matrix, in die der Anwender das gewünschte Muster zeichnen kann, und einer Reihe von Gadgets und Optionen.

Unmittelbar rechts neben der großen Zeichenmatrix befindet sich ein kleines Kontrollfeld, in dem der Anwender die in der vergrößerten Zeichenmatrix getätigten Veränderungen maßstabsgetreu dargestellt bekommt.

Dieser Editor gibt dem Anwender Möglichkeiten an die Hand, die weit über die Mosaik-Grundfunktion hinausgehen.

In einem 'Mosaik-Muster' können auch Bildelemente auftreten, die nicht in der bereits beschriebenen Weise mit einer einzelnen Farbe gefüllt werden. Ein Beispiel: Sie rufen den <Mosaik Shape Editor> auf und zeichnen in die Entwurfmatrix ein Rechteck. Nun wird der Editor verlassen und die <Mosaik>-Option angewandt.

Sie werden feststellen, daß dieses Mal das Bild nicht mehr nur ausschließlich aus groben, einfarbigen Quadraten besteht, sondern daß eine einfarbige Matrix aus farbigen Rahmen über das Bild gelegt wurde, unter dieser Matrix aber nach wie vor das eigentliche Bild vorhanden ist. Unmittelbar neben dem kleinen Kontrollbildschirm der Editmatrix befindet sich das <Blank>-Gadget, mit dem der Anwender bestimmen kann, welche Pixel als Mosaikflächen erscheinen sollen und welche in ihrem ursprünglichen Zustand bleiben. Wird <Blank> aktiviert, invertiert sich dadurch die Wirkungsweise der <Mosaik>-Funktion.

Unmittelbar unter dem <Blank>-Gadget findet der Anwender eine Reihe von Eingabefeldern, die wie folgt bezeichnet sind: <Shape>, <Linie>, <Offset>, <Flip> und <Reverse>. <Shape> und <Linie> stehen für die horizontale und vertikale Ausdehnung des Mosaikmusters, das gerade bearbeitet wird. Sollte ein Mosaikmuster größer als die maximal darstellbare Editiermatrix sein, kann der Anwender durch das Benutzen der Pfeiltasten unter der Editiermatrix dieselbe in alle vier Richtungen bewegen. Die Darstellung in dem kleinen Demofenster folgt automatisch.

Normal stellt die Mosaikstruktur, die Butcher erzeugt, eine Matrix dar. Dies kann jedoch über den Punkt <Offset> verändert werden. Solange der unter <Offset> dargestellte Wert mit dem <Shape>-Wert identisch ist, bleibt die Ausrichtung der waagrechten und senkrechten Mosaikteile unbehelligt, wird ein abweichender Wert eingegeben, werden die Mosaikteile Reihe für Reihe versetzt zueinander angeordnet.

<Flip> kippt die in der Editormatrix aktuelle Darstellung senkrecht (oben wird unten)

<Rotate> dreht die Darstellung in der Editormatrix um 90 Grad.

<Reverse> stellt die aktuelle Darstellung in der Editormatrix seitenverkehrt dar.

<Invert> verkehrt den Inhalt der Editormatrix in eine negative Darstellung.

<Clear> löscht den aktuellen Inhalt der Editormatrix.

<Restore>, <Keep> und <Cancel> haben dieselbe Funktion wie schon mehrfach in diesem Beitrag beschrieben.

Der <Mosaik Shape Editor>, ist der letzte Punkt des <Shape Mosaic>-Untermenüs.

<Slice Plane>, der nächste Punkt des <Process>-Menüs, dient dem Separieren einzelner Bitplanes eines Bildes. Wird der Punkt aktiviert, gelangt der Anwender in ein kleines Window. Die ersten sechs numerierten Felder stehen für die sechs maximal möglichen Bitplanes des Amiga, die hier durch Anklicken ein- beziehungsweise ausgeschaltet werden können. <False> schaltet auf eine schwarzweiße Darstellung um, in der die Unterschiede beim Zu- und Wegschalten einzelner Bitplanes deutlicher sichtbar sind. <View> blendet das <Slice>-Window aus.

<Restore>, <Keep> und <Chancel> haben dieselbe Funktion wie im <Filter>- und <Edge>-Window.

<Density Slice>

Durch den Menüpunkt <Density Slice> gelangt der Anwender in das <Density Slice>-Window. Unmittelbar nach dem Aktivieren beginnt die Funktion alle Bildpunkte zu zählen. Die ermittelten Werte werden dem Benutzer in Form eines Stabdiagrammes über der Palettendarstellung in der linken Hälfte des Windows ausgegeben.

<Density Slice> dient der Manipulation beliebiger Pixel einer oder mehrerer Farben. Während des Zählens der Bildpunkte ermittelt Butcher außerdem für jedes Farbregister einen Farbintensitätswert. Nach diesem Wert sortiert Butcher die Register nach Intensitätswert in aufsteigender Reihenfolge. Unmittelbar unter der Palettendarstellung befindet sich ein Schieberegler mit dem die 'Schwarzschwelle' eingestellt werden kann. Alle Farben, deren Intensitätswerte niedriger ist als der Intensitätswert des Registers, auf das die 'Schwarzschwelle' eingestellt ist, werden schwarz.

Mit <M1> und dem Schieberegler unter der Palettendarstellung wird die Schwarzschwelle bestimmt. <M2> steht für eine zweite Schwelle die auf dieselbe Weise wie die Schwarzschwelle gesetzt wird. Unter <M1> und <M2> befindet sich ein Schalter, der zwei Zustände annehmen kann: <Color> und <Mono>. Der Prozeß des Ausblendens von Farben mit niedrigen Intensitätswerten kann entweder an dem vollfarbigen Orginalbild mit 32 verschiedenen Farben erfolgen oder aber in einem monochromen Modus. Dies wird mit <Color> bzw. <Mono> eingestellt. <Isolate> aktiviert die Funktion.

<Rev> kehrt die Reihenfolge der Farbregister um. <Lock> verriegelt die ersten beiden Farbregister. <View> blendet das <Density Slice Window>

aus. Ein Druck auf die rechte Maustaste bringt das Window wieder auf den Monitor. <Restore>, <Keep> und <Cancel> haben auch in diesem Window die bereits beschriebenen Funktionen.
Sollten sich die Farben im Laufe der Arbeit so verstellen, daß ein Weiterarbeiten nicht mehr möglich ist, bringt die Taste 'R' das Bild in seinem ursprünglichen Zustand wieder auf den Monitor.

<Segment Color>
Auch hinter der nächsten Funktion des <Process>-Menüs steckt wieder ein Window, in dem der Anwender mit sechs Schiebereglern konfrontiert wird. In diesem Window kann der Anwender nun nach Belieben jedes Farbregister einzeln ein- beziehungsweise wieder ausschalten. Dazu brauchen die betreffenden Farben nur auf der Palettendarstellung in diesem Window angeklickt zu werden. <View> schaltet wie üblich kurzzeitig das Window aus. Es kann durch das Betätigen der rechten Maustaste wieder auf den Bildschirm gebracht werden.
<Color> beziehungsweise <Mono> (zwei Zustände eines Schalters) schaltet zwischen farbiger und monochromer Darstellung um, während <Gray> im aktiven Zustand dafür sorgt, daß eventuell in der Palette enthaltene Grautöne nicht verändert werden können. <Restore>, <Keep> und <Cancel> haben auch in diesem Window die bereits beschriebenen Funktionen. Unter <Screen> findet der Butcherbenutzer ein kleines Untermenü mit den Optionen <Load>, <Save> und <Edit>.
<Load> und <Save> öffnen die bekannten Laden- und Speichern-Requester, die schon an anderer Stelle ausführlich beschrieben wurden. Der Menüpunkt <Edit> öffnet das <Density>-Window.

Grundsätzlich geht es bei dieser Funktion um eine Umwandlung von farbigen Bildern in schwarzweiße Rastervorlagen. Das besondere hierbei ist das Raster, denn die Bilder werden nicht etwa einfach in verschiedene Grauwerte konvertiert, sondern tatsächlich in Raster umgesetzt wie beim Zeitschriftendruck. Die Rasterstruktur kann im <Density>- Window editiert werden und über die vorhergehenden Menüpunkte <Load> und <Save> eingeladen beziehungsweise abgespeichert werden.

Die Zahlenfelder im <Density>-Window stehen für Rasterstrukturen und können nach Belieben manipuliert werden. Durch die beiden Pfeilfelder neben den Zahlenfeldern kann der Anwender zwischen den einzelnen Seiten der Rasterdaten (enthalten in den Zahlenfeldern) blättern. <Screen> rastert das aktuelle Bild nach den gegebenen Werten auf. <Rotate> dreht die Werte in den Zahlenfeldern um 90 Grad. Ist die Option <Cpy> (Copy) akti-

Da geht's lang

Auflösung zu den Übungen aus Teil 1

<Spare Picture>
Laden Sie ein beliebiges Bild ein, und kopieren Sie es in den <Spare>-Bildschirm. Sie können dies entweder durch die Aktivierung der Option im <Projekt>-Menü tun oder mittels der Tastenkombination Amiga + P. Manipulieren Sie das aktuell sichtbare Bild mittels irgendeiner Butcher-Funktion. Nun können die beiden Bildinhalte mittels der drei Funktionen <Merge in Front>, <Merge in Back> oder <Mix Pages> vermischt werden. <Draw>
Öffnen Sie das <Projekt>-Menü und wählen Sie dort den Menüpunkt <Create Undo Map> an. Nun kann aus dem <Tools>-Menü die <Draw>-Funktion aktiviert werden.

<Clip>
Wählen Sie im <Tools>-Menü das Untermenü <Clip> an. Aus diesem Menü wird die Funktion <Make Clip> aktiviert. Nun wird auf dem eigentlichen Bild der gewünschte Bildausschnitt (Clip) definiert. Mit <Save Clip> aus dem <Projekt>-Menü kann nun der definierte Bildausschnitt abgespeichert werden. Laden können Sie Butcher Clips durch die <Load>-Funktion im <Projekt>-Menü.

viert, werden Änderungen im Zahlenfeld automatisch auch auf die anderen Seiten der Rasterdaten kopiert. <Prt> druckt eine Liste der aktuellen Daten aus. <Last> nimmt die letzten Änderungen im Zahlenfeld wieder zurück und <Std> (Standard) ruft die ursprünglichen Standardwerte des <Density>-Windows auf. <View> ermöglicht ein ungestörtes Betrachten des Bildes, und mit <Clip> können beliebige Bildausschnitte definiert werden. Ist ein Bildausschnitt definiert worden, wirkt die Rasterfunktion nur auf den betreffenden Ausschnitt.

<Restore>, <Keep> und <Cancel> haben auch im <Density>-Window die bereits beschriebenen Funktionen.

Auch der letzte Punkt des <Process>-Menüs, <Diffusion> wandelt das aktuelle Bild in eine gerasterte Darstellung um. Während allerdings bei der vorhergehenden Funktion Raster erzeugt wurden, wie sie im drucktechnischen Bereich bekannt sind, erzeugt <Diffusion> ein Rasterbild, das sich am ehesten mit einer Stickerei vergleichen läßt. <Diffusion> ist auch der letze Punkt des <Process>-Menüs.

<Convert>

Das letzte Menü des Butchers ist <Convert>. Hier geht es um das Konvertieren der verschiedensten Grafikformate, um das Tauschen und Mischen von Farbregistern und einiges mehr.

<Histogram>
<Histogram>ist der erste Punkt im <Convert>-Menü und vereint in sich einige sehr leistungsstarke Butcherfunktionen. Wiederum gelangt der Anwender nach dem Aktivieren dieser Option in ein Window. Neben einer Palettendarstellung finden sich hier auch noch einige weitere Funktionen. Der erste Auswahlpunkt in diesem Menü schaltet zwischen einer Palettendarstellung in der bekannten Art und Weise um zu einer Darstellung, in der die Farbregister entsprechend ihrer RGB-Anteile (Rot, Gelb, Blau) sortiert, abgebildet werden. <Pck> sorgt dafür, daß die aktuell eingestellte Farbe blinkt. Über <Use> kann der Benutzer ermitteln, wie oft eine Farbe in einem Bild vertreten ist. Nach der Aktivierung der Funktion gibt But-



Aufschlußreiche Informationen, ein Bild im Spiegel des <Histogram>-Menüs

cher den entsprechenden Wert als absolute Pixelzahl und als prozentualen Wert aus. Mit <Mrg> (Merge) können Farbregister beliebig in andere Register übernommen werden. <Ex> (Exchange) tauscht zwei Farbregister aus. <Bld> füllt alle Pixel eines Farbregisters mit der Farbe eines beliebigen anderen Registers. <Srt> sortiert alle Farben nach der Häufigkeit ihres Vorkommens im aktuellen Bild. <Cnt> zählt alle Bildpunkte. Die ermittelten Werte werden in Form eines Stabdiagramms über der Palette dargestellt. Mit <Clp> kann ein beliebiger Bildausschnitt definiert werden. Ist ein Bildausschnitt aktiviert, wirken alle Funktionen des <Histogram>-Windows nur auf den betreffenden Ausschnitt. Mit <Prt> kann sich der Anwender die aktuellen Palettenwerte ausdrucken lassen. <Rst> (Restore) nimmt alle getätigten Änderungen zurück und versetzt das Bild wieder in den Zustand, in dem es sich vor dem Aufruf des <Histogram>-Windows befand. <OK> verläßt das Window, wobei alle getätigten Änderungen übernommen werden. <Und> (Undo) nimmt die letzte Änderung, die im <Histogram>-Window getätigt wurde, zurück. <Cancel> schließt das Window.

<Quick Merge>
Hinter <Quick Merge> verbirgt sich das eben besprochene Window. Tatsächlich gelangt der Anwender auch durch diesen Punkt des <Convert>-Menüs in das <Histogram>-Window.

<Auto Merge>
<Auto Merge> kommt immer dann ins Spiel, wenn aus vielen Farben wenige gemacht werden sollen. In dem Requester, der dem Anwender nach der Aktivierung dieser Funktion präsentiert wird, kann beliebig zwischen den drei verschiedenen Paletten des Butchers umgeschaltet werden. So ist es auch möglich, beliebige Register aus allen drei Paletten zu mischen. In der unteren Hälfte des Requesters findet sich die Bezeichnung <Target> und ein Eingabefeld, das als <Maximum Colors> bezeichnet ist. Wird hier der Wert 15 eingegeben und die Aktion durch einen Klick im <OK>-Feld bestätigt, reduziert Butcher automatisch die Anzahl der verwendeten Register von 32 auf 15. Dabei mischt er jeweils die Farben zueinander, deren Farbwerte sich am meisten ähneln.

Mit <Cancel> gelangt man aus diesem Requester, ohne eine Aktion durchzuführen.

<Consolidate>
Sollten in einem Bild zufälligerweise mehrere Farbregister mit derselben Farbe belegt sein, kann der Anwender diese automatisch durch die Funktion <Consolidate> zusammenmischen lassen.

Nach der Aktion sind alle Pixel der betreffenden Farben einem Register zugeordnet. Sind keine Farbregister mit identischen Werten vorhanden, bewirkt <Consolidate> nichts.

<Sanitize>
Butcher stellt seine Menüleisten und Beschriftungen in den beiden in den ersten Farbregistern enthaltenen Farben dar. Hier finden sich bei der Voreinstellung ein dunkler und ein heller Farbton, um die Schriften lesbar zu halten. Wird durch irgendeine Aktion die Reihenfolge der Farben so verändert, daß auf den beiden ersten Registern wenig kontrastierende Farbtöne präsent sind, kann mit <Sanitize> Abhilfe geschaffen werden. Die ersten beiden Register werden mit der hellsten und dunkelsten Farbe der aktuellen Palette gefüllt.

<Flush Zero>
<Change Depth>
Durch den Menüpunkt <Change Depth> gelangt der Anwender in den <Change Depth>-Requester. Hier können durch die beiden Pfeilfelder Bitplanes zu- oder abgeschaltet werden. <OK> und <Cancel> erfüllen auch hier die bereits beschriebenen Funktionen.

Sämtliche nun noch im <Convert>-Menü enthaltenen Punkte, <To no Interlace>, <To Interlace>, <To LoRes Width>, <To HiRes Width>, <LoRes - to Ham> und <Ham to LoRes>, betreffen die Konvertierung von Bildern. Wird das <Convert>-Menü geöffnet, nachdem eine neue Grafik eingeladen wurde, sind alle ausführbaren Aktionen deutlich lesbar. Funktionen, die nicht benutzt werden können, erscheinen in der Amiga-spezifischen Geisterschrift.

Palettenende

Praxis macht den Könner. Der Butcher stellt seinen Benutzern eine Vielfalt von Möglichkeiten an die Hand, die sich vielfach nur geringfügig in ihrer Wirkung unterscheiden. Der Anwender muß nicht nur recht genau wissen, wie er den Butcher zu bedienen hat; er muß sich vielmehr auch über das angestrebte Resultat im klaren sein. Denn Tatsache bleibt, nur wer das Ziel kennt, kann einen Weg dorthin planen.

Aber nicht nur seine reine Funktionalität prädestiniert den Butcher als hilfreiches Werkzeug. Da das Programm nur zirka 100 kByte lang ist, läßt es sich wunderbar im Multitasking-Betrieb zusammen mit Programmen wie zum Beispiel Deluxe Paint benutzen. Eine Praxis, die nicht nur in der Theorie funktioniert.

(hs)

Ins Detail gegangen

Reise in die Makrowelt

Einer der Klassiker des fantastischen Films ist "Die unglaubliche Geschichte des Mr. C.", die Jack Arnold in den 50er Jahren verwirklichte.
Ein Mann gerät während einer Schiffsfahrt in einen seltsamen Niederschlag und beginnt von da an, unaufhörlich zu schrumpfen.

Diese reizvolle Idee wurde in auch für heutige Tage noch erstaunliche Bilder umgesetzt. Man erlebt den Schrumpfungsprozeß hautnah mit und bekommt einen Eindruck davon, wie sich die alltägliche Welt mit einem Male immer mehr verfremdet; und das nur aufgrund der Tatsache, daß sich die Umwelt in unglaublichem Maße vergrößert.

Diesen Effekt wollen wir in diesem Monat in der Tüftelecke zum Gegenstand des Ratens machen. Die Aufgabenstellung ist daher denkbar einfach:

Schauen Sie sich die folgenden vier Bilder genau an. Jedes Bild enthält einen Ausschnitt eines Gegenstandes, der im täglichen Leben häufig benutzt wird. Identifizieren Sie diese Gegenstände, und schreiben Sie deren Namen auf eine Postkarte.

Diese sollte an den

DMV-Verlag
Redaktion AMIGA DOS
Stichwort "Makros"
Postfach 250
3440 Eschwege

geschickt werden. Einsendeschluß ist der 14. Juli 1990. Es gilt das Datum des Poststempels.









Mitarbeiter des DMV-Verlages und des Redaktionsbüros Everts & Hagedorn sowie deren Angehörige dürfen nicht teilnehmen. Der Rechtsweg ist ausgeschlossen.

Die Preise

Zu gewinnen gibt es diesmal ein hölzernes Schachspiel und drei Exemplare des Spielprogramms "Distant Armies". Alle Preise wurden uns freundlicherweise von der Firma Casablanca zur Verfügung gestellt.

(jf)

Garry Glendown

ARP verständlich gemacht

Im Blickpunkt: die ARP-Library

Während wir im letzten Teil lediglich die Funktionen kennengelernt haben, kommen wir heute zu einer kleinen Anwendung, anhand der man den Umgang mit einigen wichtigen Teilen der Arp-Library in der Praxis erlernen kann.

Anfangen wollen wir dabei mit einer Frage: Welche Routine wird in sehr vielen Anwendungen gebraucht, ist recht kompliziert (zumindest, wenn man sie ordentlich programmieren will) und sollte trotz allem einfach in der Anwendung sein? Nein, hier ist nicht die Rede von der Programmierung einer Reset-Routine, sondern von einem Filerequester. In wieviel Variationen taucht er immer wieder auf? Allein auf den "Fishdisks" (Public-Domain-Serie) findet man mindestens fünf verschiedene als Source oder Linker-Library. Diese Requester mögen alle recht gut funktionieren und einige auch recht gut zu handhaben sein, aber sie sind doch mehr oder weniger kompliziert einzubinden. Und noch einen Nachteil haben sie: Sie sind lang, teilweise sogar sehr lang.

In der ARP-Küche

Dies ist alles recht unbefriedigend. Kochen wir doch lieber unser eigenes Süppchen. Man nehme folgende gute Zutaten:

17K Arp-library
1 Arp Linker Library
1 C-Compiler mit
1 Ausgabe dieser Zeitung
oder wahlweise:
1 Assembler mit
1 Ausgabe dieser Zeitung
sowie
1 Assembler-Kenntnisse oder
1 Mut zum Ausprobieren

Als erstes müssen wir die Arp-Library kurz im Libs:-Directory verrühren, wobei wir sie vorsichtig eintauchen lassen. Ferner müssen wir die Arp-Linker-Library (a.out) eventuell noch vorbereiten, sofern wir sie nicht fertig bekommen haben. Zu diesem Zweck begeben wir uns in das Directory, in dem die Sources für diese Library sind (eventuell 'prodocs/MANX' bzw. 'prodocs/LATTICE') und starten dort den automatischen a.out-Erzeuger (auch 'MAKE' genannt). Danach sollten dann alle nötigen Zutaten automatisch vermengt werden.

Das waren die Vorbereitungen. Kommen wir nun zum Hauptgang. Wir wollen ja einen Filerequester verwenden, also halten wir das schon mal fest:

```
struct FileRequester *freq,
*ArpAllocFreq();
char *f,*FileRequest();
```

Wichtig ist es, daß ab der Arp-Version 1.3 (Lib-Version 39.1) die Funktion ArpAllocFreq() verwendet wird, da sonst bei späteren Versionen die Kompatibilität nicht gewahrt sein könnte.

In der main()-Routine werden dann einige Einstellungen vorgenommen, die das Erscheinungsbild des Filerequesters und seine Funktion verändern. Im ersten Beispiel wird dabei nur ein ganz normaler Filerequester geöffnet, wobei als Funktion 'FRF__DoColor' verwendet wird. Damit erscheint der Requester mit einer anderen Farbverteilung.

Das zweite Beispiel ist dann schon etwas interessanter. Hier wollen wir nun das jeweilige Directory ohne die .info-Files anzeigen. Zu diesem Zweck setzen wir noch das Flag 'FRF__DoWildFunc'. Vorher wird noch der String in 'infowc' auf die Pattern-Match-Routine vorbereitet.

Zusammen mit dem DoWildFunc-Flag müssen wir allerdings noch einen weiteren Parameter angeben: nämlich die Adresse der Routine, die zum Pattern-Matching aufgerufen wird. Das heißt, daß jedesmal, wenn ein Filename eingelesen wurde, diese Routine aufgerufen wird (mit den Parametern Mask und Object). Die Routine muß dann entscheiden, ob der Name erscheint oder nicht.

Diese Aufgabe erledigt in diesem Fall die Routine 'NoInfo'. Beim Aufruf wird dabei dasselbe Flag als Maske übergeben, das den Aufruf verursacht hat, hier: FRF__DoWildFunc. Im Gegensatz zu der Original-Dokumentation, die angibt, daß in einem solchen Fall 'Object' auf einen FileInfoBlock zeigt, wird jedoch ein Zeiger auf einen AnchorPath übergeben. Dieser enthält dann den benötigten Zeiger, der wiederum den Namen des Files oder Directories enthält.

Wird festgestellt, daß es sich um einen Namen handelt, der dem Suchmuster entspricht (in diesem Fall '*.info'), so wird als Ergebnis '-1' zurückgegeben. Daraufhin zeigt der Filerequester das File oder Directory nicht an. Dagegen wird bei allen anderen Einträgen '0' zurückgegeben, sie erscheinen dann auch im Requester.

Im dritten Beispiel treiben wir's mal anders herum: Diesmal wollen wir, zum Beispiel für einen .info-File-Editor nur die Files haben, die mit '.info' enden, oder aber Directories. Zu diesem Zweck setzen wir diesmal als Funktions-Adresse 'InfoOnly'. Hier wird dann überprüft, ob das File entweder vom Typ 'Dir' ist (also fib->fib__DirEntryType>0) oder auf '.info' endet.

Im letzten Beispiel wollen wir bereits etwas früher benachrichtigt werden, nämlich bevor das Fenster angezeigt wird. Dies erreichen wir, indem wir als Flag 'FRF__NewWindFunc' setzen. Somit wird vor dem Aufruf von 'OpenWindow()' unsere Routine gestartet. Wie man sieht, wird diesmal kein AnchorPath-Zeiger übergeben, sondern ein Zeiger auf die NewWindow-Struktur, in der wir dann prompt die Höhe des Fensters ändern.

Das wär's schon. Die Programmierung der Arp-File-Requester-Funktion ist eigentlich gar nicht so schwer, und selbst komplizierte Anwendungen wie Wildcards und auch das Hinzufügen von Gadgets stellen kein großes Problem dar. Letztlich sind die so erzeugten Programme kurz (wozu auch die Verwendung der Arp-Printf-Funktion beiträgt: 'Printf' statt 'printf'): das ganze File hat, compiliert unter Manx Aztec 3.6 mit 16-Bit-Integer, weniger als 3 kByte. Das schafft man mit 'normalen' Filerequester-Routinen nie im Leben...

(vb)

Compileranweisungen:

```
.c.o:
cc $*.c +ILIB:allarp.pre
fr: fr.o
ln +q fr.o -la -lc
```

Listings

```
/*****************************************************/
/*                   ARP-Demo                        */
/*  Dieses Programm demonstriert den Umgang mit      */
/*      einigen Funktionen von ARP 1.3.              */
/*      (c) 1990 by G. Glendown & AmigaDOS           */
/*                 Sprache: Aztec C                  */
/*****************************************************/

struct FileRequester *freq,*ArpAllocFreq();
struct ArpBase *ArpBase;
struct IntuitionBase *IntuitionBase;
```

Listing: ARP-Demo

```
void *OpenLibrary();
char *f,*FileRequest();

char infowc[16]={"*.info"},noinfo[16];

VOID *NoInfo(Mask, Object)
ULONG Mask;
CPTR *Object;
{
struct AnchorPath *ap;
struct FileInfoBlock *fib;
 if (Mask==FRF_DoWildFunc) {
  ap=(struct AnchorPath *)Object;
  fib=&(ap->ap_Info);
```



Listing: ARP-Demo

```
  if (PatternMatch(noinfo,fib->fib_FileName))
   return(-1L);
  else return(0L);
 }
}

VOID *InfoOnly(Mask, Object)
ULONG Mask;
CPTR *Object;
{
struct AnchorPath *ap;
struct FileInfoBlock *fib;
 if (Mask==FRF_DoWildFunc) {
  ap=(struct AnchorPath *)Object;
  fib=&(ap->ap_Info);
  if ((PatternMatch(noinfo,fib->fib_FileName))||
   (fib->fib_DirEntryType>0))
   return(0L);
  else return(-1L);
 }
}

VOID *FinalRoutine(Mask, Object)
ULONG Mask;
CPTR *Object;
{
struct NewWindow *nw;
struct Window *win;

 switch (Mask) {
  case FRF_NewWindFunc:
   nw=(struct NewWindow *)Object;
   nw->Height+=12;
   nw->IDCMPFlags|=INACTIVEWINDOW;
/* Default IDCMP:
MOUSEBUTTONS|GADGETDOWN|GADGETUP|CLOSEWINDOW|RAWKEY|
INTUITICKS
Flags:
WINDOWDRAG|WINDOWCLOSE|SMART_REFRESH|ACTIVATE|
RMBTRAP */
   break;
  default:
   Printf("%lx\n",Mask);
```



Listing: ARP-Demo

```
 }
}

main()
{
 if (!(ArpBase=OpenLibrary("arp.library",39L))) {
  Printf("Kann Arp-Library V39 nicht oeffnen!\n");
  exit();
 }
 freq=ArpAllocFreq();
 freq->fr_FuncFlags|=FRF_DoColor;  /* 1. Test */
 freq->fr_Hail=("Ein Standard-Requester");
 DoFreq();

 PreParse(infowc,noinfo);
 freq->fr_Hail=("Keine `.info'-Files!"); /* 2. Test */
 freq->fr_FuncFlags|=FRF_DoWildFunc;
 freq->fr_Function=NoInfo;
 DoFreq();

 freq->fr_Hail=("Nur `.info'-Files & DIRs!");
                                             /* 3. Test */
 freq->fr_Function=InfoOnly;
 DoFreq();

 freq->fr_Hail=("Diesmal das Fenster groesser!");
                                             /* 4. Test */
 freq->fr_FuncFlags&=~FRF_DoWildFunc;
 freq->fr_FuncFlags|=FRF_NewWindFunc;
 freq->fr_Function=FinalRoutine;
 DoFreq();

 CloseLibrary(ArpBase);
}

DoFreq()
{
 f=FileRequest(freq);
 if (f)
  Printf("Ausgewaehlt: %s\n",f);
 else
  Printf("Nichts ausgewaehlt\n");
}
/**************** Ende von ARP-Demo *****************/
```

Listing: ARP-Demo

Fritz Cizmarov

Rund um den Drucker

Ein Druckerutility in Modula2

Ist der Drucker nun online oder offline? Diese Frage stellt sich immer wieder aufs neue. Wie man alle notwendigen Abfragen unter einen Hut bekommt, zeigt unser Utility.



Das Modul "PrinterSupport" wurde entwickelt, um die Druckerprogrammierung zu vereinfachen. Besondere Beachtung sollte die Prozedur "PrinterReady" im Implementationsteil finden. Hier wird eine Variable des Typs "prtStatus" verwendet. Dieser Typ wird gebraucht, da der Befehl "query" zwei verschiedene Ergebnisse liefern kann, je nachdem, ob der Drucker am parallelen oder seriellen Port angeschlossen ist. Entwickelt wurde das Modul mit dem M2Amiga-Compiler V3.3 unter Workbench 1.3.

Die Routine "OpenPrinter" öffnet das Printerdevice und testet, ob ein Drucker angeschlossen und empfangsbereit ist. Ist dies nicht der Fall, wird ein Requester mit einer entsprechenden Meldung geöffnet. Der Anwender hat die Wahl, ob nochmal getestet oder der Ausdruck abgebrochen werden soll. Läßt sich das Printerdevice nicht öffnen, so wird es wahrscheinlich von einem anderen Task benutzt. Man muß dann warten, bis das andere Programm seinen Ausdruck beendet hat, andernfalls hilft nur ein Reset.

Das Printerdevice wird von der Routine "ClosePrinter" wieder geschlossen. Das Printerdevice sollte nur solange geöffnet sein, wie man etwas auf dem Drucker ausgeben möchte, da sonst der Drucker für andere Programme blockiert wird. Wird das Programm abgebrochen, solange das Printerdevice geöffnet ist, dann wird dieses automatisch geschlossen.

Sollte eine der Funktionen "PCommand", "DumpRPort", "DumpScreen" oder "DumpWindow" FALSE zurückgeben, so kann mit "PrinterError" die Fehlerursache festgestellt werden. Wurde vergessen, das Printerdevice zu öffnen, oder ist das Öffnen selbst fehlgeschlagen, so bekommt man den Wert "OpenFail" zurück.

In Kürze noch einige wichtige Funktionen:

- PWrite sendet ein Zeichen an den Drucker, dies kann auch ein Steuerzeichen sowie CHR(0) sein.
- PWriteString sendet eine Zeichenkette an den Drucker.
- PWriteLn sendet einen Zeilenvorschub an den Drucker.
- PRawWrite sendet eine beliebige Bytefolge an den Drucker, es werden auch Null-Bytes gesendet (z.B. für Steuersequenzen).
- PCommand sendet ein Kommando an den Drucker. Die Kommandos findet man im Handbuch zum M2Amiga auf den Seiten 13-87 und 13-88.
- DumpRPort druckt einen beliebigen Ausschnitt aus einem Rastport.
- DumpScreen druckt den Screen aus und DumpWindow entsprechend das Window.

(vb)

Listings

```
{*****************************************************}
{*              PrinterSupport.Mod                  *}
{*          Hilfsroutinen fuer Drucker              *}
{*    (c) 1990 Fritz Cizmarov & AMIGA DOS           *}
{*    Sprache:  Modula-2                            *}
{*    Compiler: M2AMIGA v3.3                        *}
{*****************************************************}

IMPLEMENTATION MODULE PrinterSupport;

FROM SYSTEM IMPORT ADDRESS,BYTE,LONGSET,ADR;

FROM Arts IMPORT Requester,TermProcedure;

FROM Str IMPORT Length;

FROM Exec IMPORT Byte,UByte,write,IOStdReq,MsgPortPtr,
           IOStdReqPtr,CloseDevice,DoIO,OpenDevice;

FROM ExecSupport IMPORT CreatePort,DeletePort,
                        CreateExtIO,DeleteExtIO;

FROM Graphics IMPORT ColorMapPtr,RastPortPtr,
                     ViewModeSet;

FROM Intuition IMPORT WindowPtr,ScreenPtr;

FROM Parallel IMPORT pSel,paperOut,pBusy;

FROM Printer IMPORT printerName,rawWrite,prtCommand,
                    dumpRPort,query,IOPrinterPtr,Special,
                    SpecialSet,IODRPReq,IODRPReqPtr;

FROM Serial IMPORT busy,select;

IMPORT Parallel,Serial;

CONST header = "M2-PrinterSupport";

TYPE prtStatus = RECORD
                   CASE : CARDINAL OF
                     1 : parStatus : Parallel.StatusSet;
                     2 : serStatus : Serial.StatusSet;
                   END;(* CASE *)
                 END;(* RECORD *)
```

Listing 1: PrinterSupport.Mod

```

VAR prtPtr   : IOStdReqPtr;
    prtPort  : MsgPortPtr;


PROCEDURE DoPrtCommand(com: CARDINAL; adr: ADDRESS;
                                      len: LONGCARD);
BEGIN
    WITH prtPtr^ DO
      command := com;
      length  := len;
      data    := adr;
    END;(* WITH *)
    DoIO(prtPtr);
END DoPrtCommand;

PROCEDURE PrinterRequest(body: ADDRESS): BOOLEAN;
BEGIN
  RETURN Requester(ADR(header),body,ADR("RETRY"),
                   ADR("CANCEL"));
END PrinterRequest;

PROCEDURE PrinterReady(): BOOLEAN;
VAR status : prtStatus;
    erg    : BOOLEAN;
BEGIN
  IF prtPtr # NIL THEN
    DoPrtCommand(query,ADR(status),0);
    WITH status DO
      CASE prtPtr^.actual OF
        1 : erg := NOT (pBusy IN parStatus)
                   AND (pSel IN parStatus);
        2 : erg := NOT (busy IN serStatus)
                   AND (select IN serStatus);
      END;(* CASE *)
    END;(* WITH *)
   ELSE
    erg := FALSE;
  END;(* IF *)
  RETURN erg;
END PrinterReady;

PROCEDURE OpenPrinter(): BOOLEAN;
VAR erg : BOOLEAN;
BEGIN
  prtPort := CreatePort(NIL,0);
  IF prtPort # NIL THEN
    prtPtr := CreateExtIO(prtPort,SIZE(IODRPReq));
```

Listing 1: PrinterSupport.Mod

```
    IF prtPtr # NIL THEN
      REPEAT
        OpenDevice(ADR(printerName),0,prtPtr,
                   LONGSET{0});
      UNTIL (prtPtr^.error = 0) OR
            (NOT PrinterRequest(ADR("Can't open
             Printerdevice!")));
      IF prtPtr^.error = 0 THEN
        REPEAT
          erg := PrinterReady();
        UNTIL erg OR NOT PrinterRequest(ADR("Printer
                         NOT Ready!"));
        IF NOT erg THEN
          CloseDevice(prtPtr);
        END;(* IF *)
      END;(* IF *)
     ELSE
      erg := Requester(ADR(header),ADR("Can't create
                       IORequest!"),NIL,ADR("CANCEL"));
    END;(* IF *)
   ELSE
    erg := Requester(ADR(header),ADR("Can't create
                     Port!"),NIL,ADR("CANCEL"));
  END;(* IF *)
  IF NOT erg THEN
    IF prtPtr # NIL THEN DeleteExtIO(prtPtr) END;
    IF prtPort # NIL THEN DeletePort(prtPort) END;
    prtPtr := NIL;
    prtPort := NIL;
  END;(* IF *)
  RETURN erg;
END OpenPrinter;

PROCEDURE ClosePrinter;
BEGIN
  IF prtPtr # NIL THEN
    CloseDevice(prtPtr);
    DeleteExtIO(prtPtr);
    prtPtr := NIL;
  END;(* IF *)
  IF prtPort # NIL THEN
    DeletePort(prtPort);
    prtPort := NIL;
  END;(* IF *)
END ClosePrinter;

PROCEDURE PrinterError(): Byte;
BEGIN
```
Listing 1: PrinterSupport.Mod




```
  IF prtPtr # NIL THEN
    RETURN prtPtr^.error;
   ELSE
    RETURN openFail;
  END;(* IF *)
END PrinterError;

PROCEDURE PWrite(ch: CHAR);
BEGIN
  IF prtPtr # NIL THEN
    DoPrtCommand(write,ADR(ch),1);
  END;(* IF *)
END PWrite;

PROCEDURE PWriteString(str: ARRAY OF CHAR);
BEGIN
  IF prtPtr # NIL THEN
    DoPrtCommand(write,ADR(str),Length(str));
  END;(* IF *)
END PWriteString;

PROCEDURE PWriteLn;
BEGIN
  PWrite(12C);
END PWriteLn;

PROCEDURE PRawWrite(str: ARRAY OF BYTE; len: INTEGER);
BEGIN
  IF prtPtr # NIL THEN
    DoPrtCommand(rawWrite,ADR(str),len);
  END;(* IF *)
END PRawWrite;

PROCEDURE PCommand(com: CARDINAL; p0,p1,p2,p3: UByte)
                  : BOOLEAN;
VAR comPtr : IOPrinterPtr;
BEGIN
  IF prtPtr # NIL THEN
    comPtr := ADDRESS(prtPtr);
    comPtr^.command := prtCommand;
    comPtr^.prtCommand := com;
    comPtr^.parm0 := p0;
    comPtr^.parm1 := p1;
    comPtr^.parm2 := p2;
    comPtr^.parm3 := p3;
    DoIO(comPtr);
    RETURN prtPtr^.error = noErr;
```
Listing 1: PrinterSupport.Mod



```
    ELSE
      RETURN FALSE;
    END;(* IF *)
END PCommand;

PROCEDURE DumpRPort(rPort: RastPortPtr;
          cMap: ColorMapPtr; mod: ViewModeSet;
          x,y,w,h: CARDINAL; destW,destH: LONGINT;
          spec: SpecialSet): BOOLEAN;
VAR ioDRPReqPtr : IODRPReqPtr;
BEGIN
  IF prtPtr # NIL THEN
    ioDRPReqPtr := ADDRESS(prtPtr);
    WITH ioDRPReqPtr^ DO
      command   := dumpRPort;
      rastPort  := rPort;
      colorMap  := cMap;
      modesHi   := 0;
      modes     := mod;
      srcX      := x;
      srcY      := y;
      srcWidth  := w;
      srcHeight := h;
      destCols  := destW;
      destRows  := destH;
      special   := spec;
    END;(* WITH *)
    DoIO(prtPtr);
    RETURN prtPtr^.error = noErr;
  ELSE
    RETURN FALSE;
  END;(* IF *)
END DumpRPort;

PROCEDURE DumpScreen(screen: ScreenPtr): BOOLEAN;
BEGIN
  IF prtPtr # NIL THEN
    WITH screen^ DO
      RETURN DumpRPort(ADR(rastPort),
             viewPort.colorMap,viewPort.modes,0,0,
             width,height,0,0,SpecialSet{fullCols,
             aspect});
    END;(* WITH *)
  ELSE
    RETURN FALSE;
  END;(* IF *)
END DumpScreen;

PROCEDURE DumpWindow(window: WindowPtr): BOOLEAN;
BEGIN
  IF prtPtr # NIL THEN
    WITH window^ DO
      WITH window^.wScreen^.viewPort DO
        RETURN DumpRPort(rPort,colorMap,modes,0,0,
               width,height,0,0,
               SpecialSet{fullCols,aspect});
      END;(* WITH *)
    END;(* WITH *)
  ELSE
    RETURN FALSE;
  END;(* IF *)
END DumpWindow;

(*------------------------------------------------*)

PROCEDURE CleanUp;
BEGIN
  ClosePrinter;
END CleanUp;

BEGIN
  prtPtr  := NIL;
  prtPort := NIL;
  TermProcedure(CleanUp);
END PrinterSupport.

(********* Ende von PrinterSupport.Mod **************)
```

Listing 1: PrinterSupport.Mod

```
(***************************************************)
(*            PrinterSupport.Def                   *)
(*        Hilfsroutinen fuer Drucker               *)
(*   (c) 1990 Fritz Cizmarov & AMIGA DOS           *)
(*        Sprache:   Modula-2                      *)
(*        Compiler:  M2AMIGA v3.3                  *)
(***************************************************)

DEFINITION MODULE PrinterSupport;

FROM SYSTEM IMPORT BYTE;

FROM Exec IMPORT Byte,UByte;

FROM Graphics IMPORT ColorMapPtr,RastPortPtr,
              ViewModeSet;

FROM Intuition IMPORT WindowPtr,ScreenPtr;

FROM Printer IMPORT SpecialSet;


(* Rueckgabewerte von PrinterError()                *)
CONST badLength        = -4;
(* IOStdReq.length enthaelt ungueltigen Wert        *)
      noCmd            = -3;
(* Ungueltiger Befehl                               *)
      aborted          = -2;
(* Befehl wurde mit AbortIO() abgebrochen           *)
      openFail         = -1;
(* Device konnte nicht geoeffnet werden             *)
      noErr            =  0;
(* Kein Fehler, alles OK!                           *)
      cancel           =  1;
(* Druckvorgang wurde unterbrochen                  *)
      notGraphics      =  2;
(* Drucker unterstuetzt keinen Grafikdruck          *)
      inverHam         =  3;
(* HAM-Bild kann nicht invertiert werden            *)
      badDimension     =  4;
(* Werte fuer Druckgroesse fehlerhaft               *)
      dimensionOvflow  =  5;
(* Werte fuer Druckgroesse zu gross                 *)
      internalMemory   =  6;
(* Zu wenig Speicher fuer interne Variablen         *)
      bufferMemory     =  7;
(* Zu wenig Speicher fuer Druckpuffer               *)
      tookControl      =  8;
(* ???                                              *)


PROCEDURE OpenPrinter(): BOOLEAN;
(* Rueckgabe: TRUE wenn Drucker bereit
Beschreibung: oeffnet Printerdevice und testet,
ob Drucker bereit                                   *)

PROCEDURE ClosePrinter;
(* Beschreibung:  Schliesst Printerdevice           *)

PROCEDURE PrinterError(): Byte;
(* Rueckgabe:       Fehlercode;
   Beschreibung:  Holt Fehlercode                   *)

PROCEDURE PWrite(ch: CHAR);
(* Eingabe:    str: Zeichen, das auf Drucker
   ausgegeben werden soll
   Beschreibung:  Gibt Zeichen auf Drucker aus      *)

PROCEDURE PWriteString(str: ARRAY OF CHAR);
(* Eingabe:  str: Zeichenkette, die auf Drucker
   ausgegeben werden soll
   Beschreibung:  Gibt Zeichenkette auf Drucker aus *)

PROCEDURE PWriteLn;
(* Beschreibung:  Sendet Linefeed an Drucker        *)

PROCEDURE PRawWrite(str: ARRAY OF BYTE; len: INTEGER);
(* Eingabe:  str: Zeichenkette, die auf Drucker
   ausgegeben werden soll
   Eingabe:  len: Anzahl der Zeichen
   Beschreibung:  Gibt Zeichenkette ohne Umwandlung
   auf Drucker aus (z.B. fuer Steuersequencen)      *)

PROCEDURE PCommand(com: CARDINAL; p0,p1,p2,p3: UByte)
          : BOOLEAN;
(* Eingabe:       com    : PrtCommand (siehe Printer)
   Eingabe:       p0..p3: Parameter
   Rueckgabe:     TRUE wenn alles OK
   Beschreibung: Sendet Drucker-Kommando an Drucker *)

PROCEDURE DumpRPort(rPort: RastPortPtr;
          cMap: ColorMapPtr; mod: ViewModeSet;
          x,y,w,h: CARDINAL; destW,destH: LONGINT;
          spec: SpecialSet): BOOLEAN;
(* Eingabe:  rPort : Zieger auf RastPort
   Eingabe:  cMap  : Zeiger auf ColorMap
   Eingabe:  mod   : Viewmodes
   Eingabe:  x,y   : Linke obere Ecke des Ausschnitts
   Eingabe:  w,h   : Breite und Hoehe des Ausschnitts
   Eingabe:  destW : Breite des Ausdrucks
   Eingabe:  destH : Hoehe des Ausdrucks
   Eingabe:  spec  : Flags fuer Ausdruck
   Rueckgabe: TRUE wenn alles OK
   Beschreibung: Gibt beliebigen Ausschnitt aus
                 Rastport auf Drucker aus           *)

PROCEDURE DumpScreen(screen: ScreenPtr): BOOLEAN;
(* Eingabe:    screen: Zeiger auf Screen
   Beschreibung:  Gibt Screen auf Drucker aus       *)

PROCEDURE DumpWindow(window: WindowPtr): BOOLEAN;
(* Eingabe:    screen: Zeiger auf Screen
   Beschreibung:  Gibt Screen auf Drucker aus       *)

END PrinterSupport.

(********** Ende von PrinterSupport.Def *************)
```

Listing 2: PrinterSupport.Def

Heiko Schmitt

Farbe, wechsle dich!

Haben Sie auch schon einmal den Wunsch verspürt, die Farben eines 'tollen' Amiga-Bildes etwas abzuändern, damit es noch besser wirkt? Wurde Ihr 'künstlerisches Schaffen' – ebenso wie meines – durch ein 'exotisches' Abspeicherformat oder eine außergewöhnliche Auflösung der Grafik gehemmt, wenn nicht gar ganz gebremst? Oder fehlte es nur am nötigen 'Kleingeld', um ein gutes Zeichenprogramm zu erstehen? In solchen Situationen wird Ihnen das Programm RGB-Change in Zukunft zur Seite stehen!

Bilder, die bearbeitet werden sollen, sind vor dem Start des Programms zu laden. Bei vielen Fraktalprogrammen, Picture-Shows und allen BASIC-Programmen ist das ohne Probleme möglich. Zusätzlich sollten Sie noch ein Screen-Save-Programm besitzen, welches Ihnen erlaubt, die geänderte Grafik auf Diskette zu sichern.

Hinweise zum Programm

Nachdem Sie das Listing abgetippt haben (...oder die DATABOX-Diskette vom freundlichen Postmann in die Hand gedrückt bekamen), speichern Sie es zusammen mit der Graphics- und Intuition.bmap sowie Amiga-BASIC auf eine Diskette. Der Screen-Saver kann dort auch gleich noch mit drauf.

Danach kann's losgehen. Bringen Sie die Grafik, deren Farben geändert werden sollen, wie gewohnt auf den Bildschirm. Kehren Sie durch Drücken der Tasten 'Commodore + N' auf die Workbench zurück, und aktivieren Sie das Programm RGB-Change. Es sucht nun systematisch alle Screens und deren Windows durch. Ein neu gefundenes Fenster wird in den Vordergrund gebracht, und der Benutzer wird gefragt, ob es sich um den gewünschten Screen oder das gewünschte Window handelt oder ob weitergesucht werden soll. Beantworten Sie die Frage durch Anklicken des entsprechenden Kästchens. Ist der zu ändernde Screen oder das zu ändernde Window gefunden, erscheint das Hauptmenü.

Die Farben können nun, wie im folgenden beschrieben, beliebig manipuliert werden. Jede Änderung wird sofort auch auf das Originalbild übertragen, damit es Ihnen möglich ist, die Wirkung sofort zu beurteilen.

Kästchen 1:
Es zeigt Ihnen die gesamte zur Verfügung stehende Farbpalette an. Die Anzahl der Farben hängt natürlich von der Zahl der Bitplanes ab, die der Screen besitzt. Falls weniger als 32 Farben zur Verfügung stehen, werden die nicht wählbaren in der Hintergrundfarbe (0) gefüllt. Intern werden jedoch immer alle 32 Farben verwaltet und (falls gewünscht) auch abgespeichert. Durch Anklicken einer der Farben wird jene zur aktuell zu bearbeitenden Farbe erklärt. Alle Manipulationen beziehen sich dann auf diese Farbe.

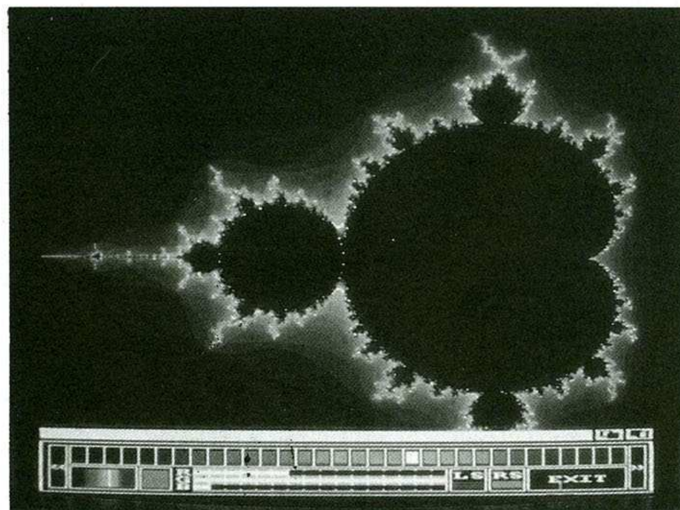
Kästchen 2:
Hier wird immer die aktuell zu bearbeitende Farbe angezeigt. Zu Beginn ist dies die Farbe Nummer 1. Durch Anklicken dieses Kästchens werden alle Kästchenumrahmungen, Schriften und andere wichtigen Informationen in der aktuellen Farbe nachgezogen. Dadurch kann im RGB-Change-Window immer ein guter Kontrast zwischen Vorder- und Hintergrund erreicht werden. Falls also nach einem Ladevorgang kein ansprechender Kontrast existiert, drücken Sie zuerst auf die Farbe in Kästchen 1, mit welcher die Umrandungen nachgezogen werden sollen, und klicken Sie danach Kästchen 2 an. Schon erscheint alles Wichtige in der von Ihnen gewählten Farbe.

Kästchen 3:
Hier werden immer die Rot-, Grün- oder Blauwerte (RGB von 0 bis 15) der aktuellen Farbe angezeigt. Durch Drücken auf eine Stelle der drei Einstell-Leisten wird der entsprechende Rot-, Grün- oder Blauwert geändert und gleich im Screen angezeigt.

Kästchen 4:
Hier wird die linke Scrollmarke angezeigt (dazu gleich mehr). Durch Anklicken dieses Kästchens wird die linke Scrollmarke auf die aktuelle Farbe gesetzt. Dadurch kann ein beliebiger, von Ihnen festgelegter Farbbereich gescrollt werden.

Kästchen 5:
Hier gilt das gleiche wie für Kästchen 4. Alle Änderungen beziehen sich in diesem Fall auf die rechte Scrollmarke.

Kästchen 6:
Der durch die Scrollmarken in Kästchen 4 und 5 festgelegte Bereich wird um ein Feld nach links bewegt (gescrollt).



RGB-Change sucht nach vorhandenen Screens und Windows und verändert auf Wunsch die Farben eines enthaltenen Bildes

Kästchen 7:
Der durch die Scrollmarken in Kästchen 4 und 5 festgelegte Bereich wird um ein Feld nach rechts bewegt (gescrollt).

Zu den Kästchen 4 bis 7 nun noch ein paar nähere Erklärungen:

Angenommen, es gibt 16 verschiedene Farben, und es sollen die Farben 5,6, 7,8,9 und 10 (also ein Teilbereich der ganzen Palette) zyklisch nach links gescrollt werden. Indem Sie die linke Scrollmarke (LS) auf die Farbe 5 setzen und die rechte (RS) auf die Farbe 10, haben Sie den gewünschten Bereich abgegrenzt. Nun klicken Sie einmal das Kästchen 6 an, und der festgelegte Ausschnitt wird nach links bewegt.

Vorher sah die Reihenfolge so aus:

Farbe:
01 02 03 04 05 06 07 08 09 10 LS RS

Nach der Änderung sieht das Ganze wie folgt aus:

Farbe:
01 02 03 04 06 07 08 09 10 05 LS RS

Mit diesen Befehlen ist es also möglich, ganze Farbblöcke an die gewünschten Stellen zu verschieben, ohne jede einzelne Farbe umzudefinieren.

Es ist natürlich auch möglich, die rechte Scrollmarke auf die Farbe 5 und die linke auf die Farbe 10 zu setzen. In diesem Fall bleiben die Farben 6, 7, 8 und 9 unangetastet, während alle anderen um ein Feld nach links verschoben werden.

Nach der Verschiebung würde sich dann folgendes Bild ergeben:

Farbe:
02 03 04 05 10 06 07 08 09 11 RS LS

Einmaliges Rechtsscrollen wird durch ein einmaliges Linksscrollen wieder aufgehoben, damit man bei Nichtgefallen der entstandenen Farbgebung den Ausgangszustand wieder herstellen kann.

Kästchen 8:
Das Programm wird (ohne Sicherheitsabfrage!) verlassen. Der Programm-Screen wird mit allen dazugehörigen Fenstern geschlossen, und der Screen, dessen Farben geändert wurden, bleibt im Vordergrund stehen. Wurde dieses Kästchen aus Versehen angeklickt, kann man durch Drücken von 'Commodore + N' auf die Arbeitsoberfläche zurückkehren und das Programm RGB-Change erneut

starten. Die schon geänderten Screen-Farben werden von RGB-Change übernommen, und es kann normal weitergearbeitet werden.

Soll der Screen, nachdem er die gewünschten Farben erhalten hat und das Programm verlassen wurde, abgespeichert werden, gehen Sie wie folgt vor:

Gehen Sie durch Drücken von 'Commodore + N' auf die Workbench, starten Sie das Screen-Save-Programm, drücken Sie 'Commodore + M' (der geänderte Screen erscheint wieder), und warten Sie, bis das Bild gespeichert wurde.

Kästchen 9:
zeigt immer die komplette Farbpalette (Farbe 0 bis 31) an.

Falls 64 Farben existieren sollten (Halfbrite-Modus), sind die Farben 32 bis 63 halb so hell wie die Farben 0 bis 31. Aus diesem Grund werden Sie hier nicht extra aufgeführt. Da im Halfbrite-Modus die Farben 32 bis 63 von den Farben 0 bis 32 abhängen, kann man Sie **nicht** alleine verändern. Es ist insofern sinnlos, sie in die Farbpalette von Kästchen 1 zu übernehmen! Im 'Hold-and-Modify'-(HAM)-Modus können Sie die Farben zwar beliebig ändern, was jedoch (in den meisten Fällen) durch eine total unterschiedliche Farbgebung in diesem Modus keinen Sinn ergibt.

Durch Anklicken dieses Kästchens (9) erscheint das Save/Load-Menü. Folgende Funktionen sind hier möglich:

1.) Filename:
Hinter dem aktuellen Filenamen erscheint ein Cursor. Mit ihm können Sie den Filenamen wie gewünscht ändern. Vergessen Sie nicht die Angabe von Laufwerksnummer und eventuellen Unterverzeichnissen. Durch Drücken von Return wird der eingegebene Filename übernommen, der Cursor verschwindet.

2.) Cancel:
Sie gelangen, ohne etwas abgespeichert oder geladen zu haben, in das Hauptmenü zurück.

3.) Save:
Die im Hauptmenü vorhandene Farbpalette wird unter dem angegebenen Filenamen abgespeichert. Die Datei besitzt Werte von 0 bis 15, die durch ein CR – CHR$(13) – getrennt sind. Sie umfaßt immer 32 Farben, selbst wenn auf dem zu ändernden Screen weniger zu sehen waren! Die Werte können im CLI durch den Type-Befehl ausgedruckt werden (ASCII-Datei). Nicht vorhandene Unterverzeichnisse oder Disketten beziehungsweise Laufwerke werden nicht erkannt und führen insofern zu Fehlermeldungen. Eventuelle Files gleichen Namens werden (radikal) überschrieben (also Vorsicht!).

4.) Load:
Eine auf Diskette gespeicherte Farbpalette wird eingeladen und gleich angezeigt. Ist kein File unter diesem Namen zu finden, führt dies (wie immer) zu den beliebten Fehlermeldungen. Falls der zu bearbeitende Screen weniger als 32 Farben anzeigen kann, werden die letzten in den Speicher gelesen, aber nicht angezeigt. Sie können jedoch weiter verwendet werden.

Auf unserer DATABOX finden Sie einige Beispielpaletten, die Files 'RGB01' bis 'RGB07' und 'WorkbenchRGB' sind sofort ladbar.

Unter 'WorkbenchRGB' finden Sie die original Workbenchfarben.

(jb)

AMIGA DOS Info

'RGB-Change ist ein Amiga-BASIC-Programm. Es benötigt zur Arbeit noch zwei '.bmap'-Dateien, die 'Intuition.bmap' und die 'graphics.bmap'. Beide '.b map'-Dateien müssen sich auf der Diskette befinden, auf der sich auch 'RGB-Change' befindet.

Das Programm arbeitet am besten mit sogenannten 'Screen-Savern' zusammen, also Programmen, die Bilder durch Tastendruck isoliert in den Speicher bringen. Die 'Screen-Saver' findet man auf vielen Public-Domain-Disketten.

Listings

```
REM  ****************************
REM  *   R G B - C h a n g e   *
REM  *   by Heiko Schmitt 1988  *
REM  *  Sprache : AmigaBASIC    *
REM  ****************************

  LIBRARY "graphics.library":LIBRARY "intuition.library"
  DIM RGB(3,31):C=1:F=0
  SCREEN 2,320,70,5,3:WINDOW 3,,(0,13)-(311,55),16,2
  SBase&=PEEKL(WINDOW(7)+46)
  CALL MoveScreen(SBase&,0,446)
  RESTORE TextDaten2

  FOR i=1 TO 3:READ x%,y%,t$
    GOSUB CursorText
  NEXT
  LINE(21,24)-(139,42),1,B:LINE(174,24)-(292,42),1,B
  Found=0:WBase&=PEEKL(SBase&+4):GOSUB SuchWindow
  IF Found=0 THEN GOTO Ende

Abfrage:
  GOSUB MausDruck
  IF WINDOW(0)<>3 OR y<26 OR y>44 THEN GOTO Abfrage
  IF x<141 AND x>23 THEN GOSUB SuchWindow
  IF Found=0 THEN Ende:ELSE:GOTO Abfrage
  IF x<176 OR x>294 THEN GOTO Abfrage:ELSE:DECLARE FUNCTION GetRGB4& LIBRARY
  Deep=PEEK(SBase&+189):CBase&=PEEKL(SBase&+48):LS=1
  IF Deep>5 THEN Deep=5
  RS=2^Deep-1:MaxRGB=RS
  WINDOW 2,,(0,13)-(311,55),16,2
  FOR Farbe%=0 TO 31
    RGB$="0000":FP%=GetRGB4&(CBase&,Farbe%)
    MID$(RGB$,5-LEN(HEX$(FP%)))=HEX$(FP%)
    RGB(1,Farbe%)=VAL("&h"+MID$(RGB$,2,1))
    RGB(2,Farbe%)=VAL("&h"+MID$(RGB$,3,1))
    RGB(3,Farbe%)=VAL("&h"+MID$(RGB$,4,1))
  NEXT Farbe%
  DIM XY(13,4)
  FOR i=0 TO RS
    PALETTE i,RGB(1,0)/15,RGB(2,0)/15,RGB(3,0)/15
  NEXT
  GOSUB WindowDraw:Na$="DF1:"
  GOSUB ColorDraw:RESTORE
  FOR i=1 TO 13
    READ XY(i,1),XY(i,2),XY(i,3),XY(i,4)
  NEXT

HauptMenue:
  GOSUB MausDruck:XY=10
  IF WINDOW(0)<>2 THEN GOTO HauptMenue
  FOR i=1 TO 9
    IF y>XY(i,2) AND y<XY(i,4) AND x>XY(i,1) AND x<XY(i,3) THEN XY=i
  NEXT
  ON XY GOTO Farb,LeSc,RiSc,LScM,RScM,RGBw,ZeFa,Ende,CoPa,HauptMenue

Ende:
  WINDOW CLOSE 2:WINDOW CLOSE 3:SCREEN CLOSE 2:LIBRARY CLOSE:END

Farb:
  IF x>MaxRGB*9+21 THEN GOTO HauptMenue:ELSE:F=INT((x-14)/9)
  GOSUB RGBdraw:GOTO HauptMenue

LeSc:
  IF LS<RS THEN Fl=LS:Fr=RS
  GOSUB LeftCycle:GOSUB RGBdraw:GOTO HauptMenue
  IF LS=RS THEN GOTO HauptMenue:ELSE:IF RS>0 THEN Fl=0:Fr=RS
  GOSUB LeftCycle
  IF LS<MaxRGB THEN Fr=MaxRGB:Fl=LS:GOSUB LeftCycle
  FOR i=1 TO 3
    SWAP RGB(i,MaxRGB),RGB(i,RS)
  NEXT
  Help=F:F=MaxRGB:GOSUB ChangeRGB
  F=RS:GOSUB ChangeRGB:F=Help:GOSUB RGBdraw:GOTO HauptMenue

RiSc:
  IF LS<RS THEN Fl=LS:Fr=RS:GOSUB RightCycle:GOSUB RGBdraw:GOTO HauptMenue
  IF LS=RS THEN HauptMenue:ELSE:IF RS>0 THEN Fl=0:Fr=RS:GOSUB RightCycle
  IF LS<MaxRGB THEN Fr=MaxRGB:Fl=LS:GOSUB RightCycle
  FOR i=1 TO 3
    SWAP RGB(i,0),RGB(i,LS)
  NEXT
  Help=F:F=LS:GOSUB ChangeRGB
  F=0:GOSUB ChangeRGB:F=Help:GOSUB RGBdraw:GOTO HauptMenue

LScM:
  LS=F:GOSUB LeftRight:GOTO HauptMenue

RScM:
  RS=F:GOSUB LeftRight:GOTO HauptMenue

RGBw:
  RGB(INT((y-19)/8),F)=INT((x-79)/8):GOSUB RGBdraw:GOSUB ChangeRGB
  GOTO HauptMenue

```

Listing: RGBChange.BAS

```
ZeFa:
  IF C=F THEN GOTO HauptMenue:ELSE:C=F:GOSUB WindowDraw:GOTO HauptMenue
CoPa:
  WINDOW OUTPUT 3:LINE(0,0)-(311,55),0,BF
  RESTORE TextDaten3
  FOR i=1 TO 4
    READ x%,y%,t$
    GOSUB CursorText
  NEXT
  LINE(91,2)-(292,20),C,B
  x%=96:y%=14:t$=Na$
  LINE(174,30)-(292,48),C,B:LINE(21,30)-(69,48),C,B:LINE(91,30)-(139,48),C,B
  GOSUB CursorText:WINDOW 3
Warten:
  GOSUB MausDruck:XY=5
  IF WINDOW(0)<>3 THEN Warten:ELSE:FOR i=10 TO 13
    IF y>XY(i,2) AND y<XY(i,4) AND x>XY(i,1) AND x<XY(i,3) THEN XY=i-9
  NEXT
  ON XY GOTO DateiName,Cancel,Speichern,Laden,Warten
  DateiName:x%=96:y%=14
  IF LEN(Na$)<24 THEN t$=Na$+"_":GOSUB CursorText
Eingabe:
  a$=INKEY$:IF a$="" THEN GOTO Eingabe:ELSE:IF a$=CHR$(8) THEN GOTO Back
  IF a$=CHR$(13) THEN IF LEN(Na$)=24 THEN Warten:ELSE:PRINT CHR$(8);:GOTO Warten
  IF a$<" " OR a$>"z" OR LEN(Na$)=24 THEN Eingabe:ELSE:IF a$=CHR$(34) THEN a$="'"
  PRINT CHR$(8);a$;:Na$=Na$+a$:IF LEN(Na$)<24 THEN PRINT"_";:GOTO Eingabe
  GOTO Eingabe
Back:
  IF LEN(Na$)=0 THEN GOTO Eingabe:ELSE:IF LEN(Na$)<24 THEN PRINT CHR$(8);
  a$=Na$:Na$=LEFT$(a$,LEN(a$)-1):PRINT CHR$(8);"_";:GOTO Eingabe
Cancel:
  WINDOW 2:GOTO HauptMenue
Speichern:
  OPEN Na$ FOR OUTPUT AS £1
  FOR i=0 TO 31
    PRINT£1,RGB(1,i):PRINT£1,RGB(2,i):PRINT£1,RGB(3,i)
  NEXT:CLOSE£1:WINDOW 2:GOTO HauptMenue
Laden:
  OPEN Na$ FOR INPUT AS £1
  FOR i=0 TO 31
    INPUT£1,RGB(1,i):INPUT£1,RGB(2,i):INPUT£1,RGB(3,i)
  NEXT
  CLOSE£1:Help=F
  FOR F=0 TO 31
    GOSUB ChangeRGB
  NEXT
  F=Help:WINDOW 2:GOSUB RGBdraw:GOTO HauptMenue
XYWerte1:
  DATA 14,4,302,24,3,4,12,51,304,4,313,51,209,27,227,51,229,27,247
  DATA 51,79,27,207,51,51,27,67,51,249,27,302,51,14,27,49,51
XYWerte2:
  DATA 95,4,294,22,176,32,294,50,23,32,71,50,93,32,141,50
TextDaten1:
  DATA 68,32,"R",68,40,"G",68,48,"B",209,35,"LS",229,35,"RS",258,40
  DATA "EXIT",2,29,"<<",303,29,">>"
  REM '<<' und '>>' entsprechen den Doppelpfeilen
  REM zu erreichen mit <ALT>+171 (<<) und <ALT>+187 (>>)
TextDaten2:
  DATA 21,14,"Ist dies das zu aendernde Fenster ?",29,36
  DATA "Weiter suchen",182,36,"Farben aendern"
TextDaten3:
  DATA 21,14,"Filename",30,42,"SAVE",100,42,"LOAD",210,42,"Cancel"
  CursorText:e%=LEN(t$):COLOR C,0:CALL Move(WINDOW(8),x%,y%)
  CALL Text(WINDOW(8),SADD(t$),e%):RETURN
MausDruck:
  WHILE MOUSE(0)<>0:WEND
  WHILE MOUSE(0)=0:WEND
  x=MOUSE(1):y=MOUSE(2)
  RETURN
ShowWindow:
```

Listing: RGBChange.BAS

```
  Found=1:CALL ScreenToFront(SBase&)
  IF WBase&>0 THEN CALL WindowToFront(WBase&):WINDOW 3:RETURN
SuchWindow:
  WBase&=PEEKL(WBase&):IF WBase&>0 THEN ShowWindow
SuchScreen:
  SBase&=PEEKL(SBase&):IF SBase&=0 THEN Found=0
  RETURN
  WBase&=PEEKL(SBase&+4):GOTO ShowWindow
FarbClear:
  FOR i=1 TO MaxRGB
    PALETTE i,RGB(1,0)/15,RGB(2,0)/15,RGB(3,0)/15
  NEXT
  RETURN
RGBdraw:
  FOR i=1 TO 3
    LINE(78,19+i*8)-(85+RGB(i,F)*8,23+i*8),C,BF
    IF RGB(i,F)<15 THEN LINE(86+RGB(i,F)*8,19+i*8)-(204,23+i*8),0,BF
  NEXT
  LINE(51,28)-(63,46),F,BF
  RETURN
WindowDraw:
  LINE(12,25)-(47,49),C,B:LINE(12,2)-(300,22),C,B:RESTORE TextDaten1
  FOR i=0 TO 3
    LINE(77,25+i*8)-(205,25+i*8),C
  NEXT
  FOR i=0 TO 31
    LINE(i*9+21,2)-(i*9+21,22),C
  NEXT
  FOR i=1 TO 8
    READ x%,y%,t$
    GOSUB CursorText
  NEXT
  LINE(1,2)-(10,49),C,B:LINE(49,25)-(65,49),C,B
  FOR i=77 TO 205 STEP 8
    LINE(i,25)-(i,49),C
  NEXT
  FOR i=0 TO 2
    LINE(78,i*8+27)-(204,i*8+31),0,BF
  NEXT
  LINE(207,25)-(225,49),C,B:LINE(227,25)-(245,49),C,B
  LINE(247,25)-(300,49),C,B:LINE(302,2)-(311,49),C,B
  GOTO RGBdraw
ColorDraw:
  GOSUB RGBdraw
  FOR i=0 TO MaxRGB
    LINE(i*9+14,6)-(i*9+19,18),i,BF:LINE(14+i,28)-(14+i,46),i
  NEXT
  GOSUB LeftRight
  FOR i=0 TO MaxRGB
    PALETTE i,RGB(1,i)/15,RGB(2,i)/15,RGB(3,i)/15
  NEXT
  RETURN
LeftRight:
  LINE(209,39)-(223,45),LS,BF:LINE(229,39)-(243,45),RS,BF
  RETURN
ChangeRGB:
  R%=RGB(1,F):G%=RGB(2,F):B%=RGB(3,F)
  PALETTE F,R%/15,G%/15,B%/15
  F%=F
  CALL SetRGB4CM(CBase&,F%,R%,G%,B%):CALL RemakeDisplay
  RETURN
LeftCycle:
  Help=F:R=RGB(1,Fl):G=RGB(2,Fl):B=RGB(3,Fl)
  FOR F=Fl TO Fr-1
    FOR j=1 TO 3
      RGB(j,F)=RGB(j,F+1)
    NEXT
    GOSUB ChangeRGB
  NEXT
  F=Fr:RGB(1,Fr)=R:RGB(2,Fr)=G:RGB(3,Fr)=B:GOSUB ChangeRGB:F=Help
  RETURN
RightCycle:
  Help=F:R=RGB(1,Fr):G=RGB(2,Fr):B=RGB(3,Fr)
  FOR F=Fr TO Fl+1 STEP -1
    FOR j=1 TO 3
      RGB(j,F)=RGB(j,F-1)
    NEXT
    GOSUB ChangeRGB
  NEXT
  F=Fl:RGB(1,Fl)=R:RGB(2,Fl)=G:RGB(3,Fl)=B:GOSUB ChangeRGB:F=Help
  RETURN
```



Listing: RGBChange.BAS



Michael Herbst

Vektorverbieger

Die Sache mit dem Chip-RAM

Haben Sie Ihren Amiga auf 1 MByte aufgerüstet und den neuen 'Fat Agnus' eingebaut oder nennen Sie den Amiga 2000HD Ihr eigen? Laufen Ihre Programme jetzt nicht mehr, obwohl sie mit den Amigas der älteren Generation einwandfrei liefen?

Die Ursache dafür könnte in der Tatsache liegen, daß die Rückgabewerte einiger Betriebssystemfunktionen ignoriert wurden. Das Problem dabei liegt beim Aufruf der Allocmem-Routine der exec.library, wenn als Bedingung im Register D1 MEMF__FAST (= $4) übergeben wird. Die Routine gibt dann natürlich NULL (= Fehler) zurück, wenn ein Amiga zwar 1 MByte RAM besitzt, dies jedoch ausschließlich Chip-RAM ist. Wird diese Rückgabe ignoriert, kommt es mit Sicherheit zum Absturz.

Das Problem dürfte ab jetzt der Vergangenheit angehören, wenn das Programm Patchallocmem bei besagten Amigas in der Startup-Sequenz aufgerufen wird. Dieses Programm verbiegt den Allocmem-Vektor auf eine eigene Routine, die das Register D1 mit der Bedingung MEMF__FAST (falls eingetragen) immer in MEMF__CHIP (= $2) abändert. Danach erst wird die alte Allocmem-Routine aufgerufen. Damit sollten nun die Programme, die früher abstürzten, nun wieder lauffähig sein.

Patchallocmem wurde mit dem DevPac Assembler V2.0 geschrieben. Eine Umsetzung auf andere Assembler ist bis auf die bekannten Feinheiten (beispielsweise ist cnop 0,2 bei manchen Assemblern durch even oder align zu ersetzen) kein Problem. Anfangs wird getestet, ob der Allocmem-Vektor bereits verbogen ist (Programm wurde schon einmal aufgerufen). Bei positivem Ergebnis wird eine entsprechende Meldung ausgegeben und das Programm beendet. Ist dies nicht der Fall, wird der Allocem-Vektor auf eine eigene Routine gelegt, eine Meldung darüber ausgegeben und das Programm verlassen. Sollten während des Programmablaufs Fehler auftreten, werden diese durch Ändern der Hintergrundfarbe angezeigt, wobei gilt:

- Rot = Fehler beim Funktionsaufruf __LVOSetFunction
- Gelb = Fehler bei der Speicherbeschaffung mittels __LVOAllocmem.

Es empfiehlt sich ein Aufruf in der Startup-Sequenz mit Patchallocmem oder Patchallocmem >NIL

Das Programm Patchallocmem darf nur vom CLI/Shell, jedoch nicht von der Workbench aus gestartet werden. Viel Spaß beim Vektorverbiegen.

(vb)

Listings

```
;*****************************************************************
;*                     PatchAllocMem                             *
;*  Patch-Programm fuer Amigas mit 1 MB Chip-Memory              *
;*                   ohne Fast-Memory                            *
;*          (c) 1990 Michael Herbst & AMIGA DOS                  *
;*   Sprache:  DevPac Assembler V2.0                             *
;*****************************************************************

opt l-,c+

;* -------------------
;* Ein paar Konstanten
;* -------------------

;* Library-Vector-Offsets exec.library

SetFunction     equ -420
AllocMem        equ -198
FreeMem         equ -210
Forbid          equ -132
Permit          equ -138
OpenLibrary     equ -552
```



Listing: Patchallocmem

```
CloseLibrary    equ -414

;* Library-Vector-Offsets dos.library

Output          equ -60
Write           equ -48

;* ----------------------
;* Das eigentliche Programm
;* ----------------------

Start:
     move.l     $4.w,a6          ;ExecBase nach a6
     jsr        Forbid(a6)       ;Multitasking verbieten
     move.l     #Memory-NewAllocMem,d1
     ;*** überpruefen, ob der AllocMem-Vektor
     move.l     -196(a6),d0
     ;*** evtl. schon von uns veraendert ist
     move.l     d0,a0
     move.l     (a0,d1.l),d1
     cmp.l      d0,d1
     beq.s      set               ;Er ist schon verbogen!
     moveq.l    #NewAllocMemEnd-NewAllocMem,d0
     ;*** Da er noch nicht neu gesetzt ist,
     move.l     #$20002,d1
```

Listing: Patchallocmem

```
    ;*** Speicher fuer die neue (kurze)
    jsr        AllocMem(a6)         ;Routine beschaffen
    tst.l      d0
    beq        fehlerallocmem
    ;*** Speicher konnte nicht belegt werden(?)
    move.l     d0,Memory      ;ansonsten Adresse merken
    move.l     #NewAllocMemEnd-NewAllocMem-1,d0
    ;*** Jetzt unsere Routine an die belegte Adresse
    lea        NewAllocMem(pc),a0          ; kopieren
    move.l     Memory,a1

CopyLoop:
    move.b     (a0)+,(a1)+
    dbra       d0,CopyLoop
    ;*** Nun ist die Routine komplett kopiert

    move.l     $4.w,a6
    ;*** Jetzt gilt es, unsere Funktion als neue
    move.l     a6,a1
    ;*** _LVOAllocMem-Routine einzutragen
    move.l     Memory,d0
    ;*** In Memory steht der Beginn der Routine
    move.l     #-198,a0
    ;*** _LVOAllocMem hat den Offset -198
    jsr        SetFunction(a6)
    ;*** Routine wird als neue Funktion eingetragen
    tst.l      d0                 ;Fehler unterlaufen (?)
    beq.s      nosetfunction
    ;*** nicht weitermachen (Speicher freigeben)
    move.l     Memory,a0
    ;*** alter Vektor muss an die entsprechende Stelle
    move.l     #jmp-NewAllocMem,d1
    ;*** eingetragen werden, damit nach dem Veraendern
    ;*** des Registers D1
    move.l     d0,2(a0,d1.l)
    ;*** in alte Routine eingesprungen werden kann!
    bra.s      text               ;Eine Meldung muss sein

set:
    move.w     #$F,alreadyset
    ;*** Vektor ist schon von uns veraendert!

text:
    lea        dosname(pc),a1
    ;*** Wir geben einen kurzen Text im CLI/Shell aus
    moveq      #0,d0
    ;*** Dazu muss die dos.library geoeffnet werden
    jsr        OpenLibrary(a6)
    tst.l      d0                         ;Fehler (???)
    beq.s      BackToCLI          ;Dann zurueck ins CLI,
    move.l     d0,a6
    ;*** ansonsten Output-Kanal holen
    ;*** (ist das CLI/Shell-Fenster)
    jsr        Output(a6)
    move.l     d0,d1
    tst.w      alreadyset
    ;*** Je nach Status von alreadyset Text ausgeben,
    ;*** dass der Vektor
    bne.s      AlreadyPatched
    ;*** schon veraendert ist/gerade veraendert wurde
    move.l     #PatchEnd-Patch,d3
    move.l     #Patch,d2

schreibe:
    jsr        Write(a6)
    ;*** Nun Text im CLI/Shell ausgeben
    move.l     a6,a1              ;DosBase wieder nach a1
    move.l     $4.w,a6                ;ExecBase nach a6
    jsr        CloseLibrary(a6)
    ;*** dos.library wieder schliessen

BackToCLI:
    jsr        Permit(a6)
    ;*** Multitasking wieder erlauben
```

Listing: Patchallocmem

```
    moveq      #0,d0              ;RETURN_OK zurueckgeben
    rts                           ;und Programm beenden

AlreadyPatched:
    move.l     #AlreadyPatchedTextEnd-AlreadyPatchedText,d3
    move.l     #AlreadyPatchedText,d2
    bra.s      schreibe

NewAllocMem:
    bclr.l     #2,d1
    bset.l     #1,d1

jmp:
    jmp        $fc0000
    ;*** Hier wird spaeter die richtige Adresse
    ;*** eingetragen (z.B. c03a0 usw.)

Memory:
    dc.l       0

NewAllocMemEnd:

nosetfunction:
    move.l     Memory,a1
    ;*** Die Funktion SetFunction ergab einen Fehler,
    moveq.l    #NewAllocMemEnd-NewAllocMem,d0
    ;*** deshalb kann der belegte Speicher wieder
    jsr        FreeMem(a6)            ;freigegeben werden
    move       #$0F00,d1
   ;*** Warnung: Bildschirmhintergrund kurz rot werden
    bra.s      color

fehlerallocmem:
    move       #$0FF0,d1  ;kein Speicher belegt, daher
    ;*** eine Warnmeldung durch gelbe Hintergrundfarbe

color:
    move.l     #$FFFF,d0

colorloop:
    move       d1,$DFF180
    ;*** Farbwert in COLOR00 eintragen
    dbra       d0,colorloop                  ;(65536-mal)
    bra.s      BackToCLI               ;Programm beenden

;* --------------
;* Ein paar Daten
;* --------------

dosname:
    dc.b       "dos.library",0
    cnop       0,2

Patch:
    dc.b       10,13,"AllocMem patched.",10,13
    dc.b       "Requirement MEMF_FAST will be changed to MEMF_CHIP.",10,13
    dc.b       155,"33;3m(c) 1990 by MIC & AmigaDOS.",155,"0m",10,13
    cnop       0,2

PatchEnd:
AlreadyPatchedText:
    dc.b       10,13,"AllocMem already patched.",10,13
    cnop       0,2

AlreadyPatchedTextEnd:
alreadyset:
    dc.w       0

;**************** Ende von PatchAllocMem *************
```



Listing: Patchallocmem

Ulf Petersen

Die Blitter-Ecke

Teil 5 : Linien mit dem Blitter

Wohl jeder gerät ins Staunen, wenn er Spiele mit schneller Vektorgrafik sieht. Weshalb das so schnell geschehen kann, das soll in diesem Kursteil erläutert werden.

Bereits im ersten Kursteil wurde erwähnt, daß der Blitter die Fähigkeit besitzt, schnell Daten zu kopieren, und außerdem in der Lage ist, Linien zu zeichnen. Daß dies mit einer atemberaubenden Geschwindigkeit von einer Million Punkten pro Sekunde geschieht, dürfte wohl nach den bisherigen Kursteilen niemanden mehr erstaunen, denn schon das Kopieren von Daten ging rasend schnell. Was eine Linie ist, dürfte bekannt sein, nämlich die Verbindung zweier gewählter Punkte durch eine Gerade. Diese Geraden dürfen für den Blitter eine maximale Länge von 1024 Punkten besitzen. Ob es beim neuen BIG-Agnus, in dem ja die Blitter-Register stecken, anders ist, ist momentan noch etwas unklar. Um aber den Standard zu wahren, sollten nur maximal 1024 Punkte verwendet werden.

Doch zurück zum Thema. Es genügt leider nicht, wenn man dem Blitter einfach den Anfangs- und den Endpunkt angibt. Vielmehr sind noch eine ganze Reihe weiterer Schritte notwendig, denn man muß die Koordinaten wieder Blitter-gerecht definieren. Das Koordinatenkreuz des Blitters ist nämlich in 8 Teile, die sogenannten Oktanten, unterteilt (siehe Abb. 1). Für den Blitter ist es nun notwendig zu wissen, in welchem Oktanten der Endpunkt (!) der Linie liegt. Dieser Oktant kann ganz einfach durch drei logische Vergleiche ermittelt werden. Diese Vergleiche bestehen daraus, die Koordinaten des Start- und des End-Punktes gegenüberzustellen. Tabelle 1 gibt Ihnen eine Übersicht über die notwendigen Vergleiche.

DeltaX und DeltaY bilden die Beträge der beiden Koordinatenachsen. Somit ist DeltaX der Betrag (!) der Differenz *X2-X1* und DeltaY der Betrag der Differenz *Y2-Y1*. Hat man nun den betreffenden Oktanten herausgefunden (bei mehreren Möglichkeiten ist die Wahl egal), schreibt man die Code-Bits und nicht den Oktantenwert in die betreffenden Stellen des BLTCON1-Registers. Die Code-Bits setzen sich dabei aus den folgenden Einzelbits zusammen (in der richtigen Reihenfolge):

SUL (Sometimes up or left),
SUD (Sometimes Up or Down)

und

AUL (Always Up or Left).

An dieser Stelle werfen Sie bitte einen Blick auf das BLTCON1-Register. Zwar hatten wir es schon in einem früheren Kursteil vorgestellt, aber für den Linienmodus haben die einzelnen Bits völlig neue Funktionen bekommen, die Sie der Tabelle 2 entnehmen können.

Neben den Code-Bits des Oktanten müssen wir dem Blitter noch die Steigung definieren, und wie sich jeder denken kann, wieder einmal Blitter-gerecht. Dazu muß man erst einmal den DeltaX- und den DeltaY-Wert miteinander vergleichen. Das kleinere Delta beziehungsweise den kleineren Wert nennen wir nun KDelta (Kleineres Delta) und den größeren Wert GDelta (Größeres Delta). Aus diesen beiden Deltas müssen nun drei Werte errechnet werden. Die Gleichungen dazu lauten folgendermaßen:

*1) 2*KDelta*
*2) 2*KDelta-GDelta*
*3) 2*KDelta-2*GDelta*

Diese Werte müssen in bestimmte Blitter-Register geschrieben werden, die gleich noch erläutert werden. Für den Fall, daß 2*KDelta < GDelta sein sollte, muß außerdem noch unbedingt das SIGN-Bit im BLTCON1-Register gesetzt werden. Ansonsten wird es wie immer auf LOW (0) gesetzt.

Linienmuster

Selbstverständlich bietet der Blitter auch beim Zeichnen von Linien die Möglichkeit, ein Muster darzustellen. In diesem Falle wird in das Register BLTBDAT die gewünschte Maske geschrieben. Da dieses Register wie üblich 16 Bits umfaßt, wiederholt sich das angegebene Muster alle 16 Bildschirmpunkte, wobei die Linienpunkte selbstverständlich die absolute Priorität über den Hintergrund besitzen. Sollte ein Hintergrundbit also auf 1 und das Maskenbit auf 0 stehen, so wird das betreffende Bit wegen der Priorität nicht gesetzt.

Begrenzung

Im vierten Teil unserer Blitter-Ecke wurde bereits erläutert, daß die Begrenzungen bei Flächen immer nur einen Pixel breit sein dürfen, da sonst der Füllalgorithmus des Blitters durcheinandergerät.

Glücklicherweise hat man diesen Umstand auch beim Blitter berücksichtigt, wenn man mit ihm Begrenzungslinien zeichnen will. Kommt das also in Frage, so muß das SIGN-Bit im BLTCON1-Register auf 1 gesetzt werden, damit der Blitter automatisch nur einen Punkt pro Zeile bei der aktuellen Linie setzt. Zum genaueren Verständnis schauen Sie sich bitte Tabelle 3 an.

Weitere Register

An dieser Stelle werde ich noch auf die weiteren Register eingehen, die zum Zeichnen von Linien benötigt werden. Sie sind Ihnen zwar schon vom Namen her bekannt, jedoch haben sich Ihre Funktionen im Linienmodus geändert.

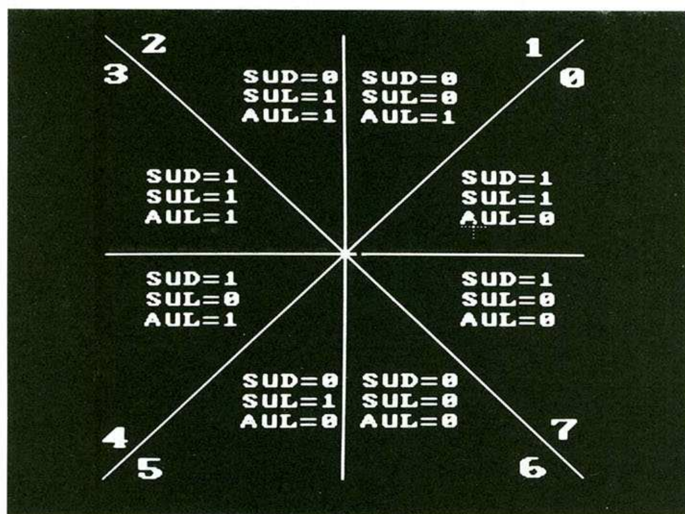


Bild 1. Das Koordinaten-System des Blitters – die Oktanten

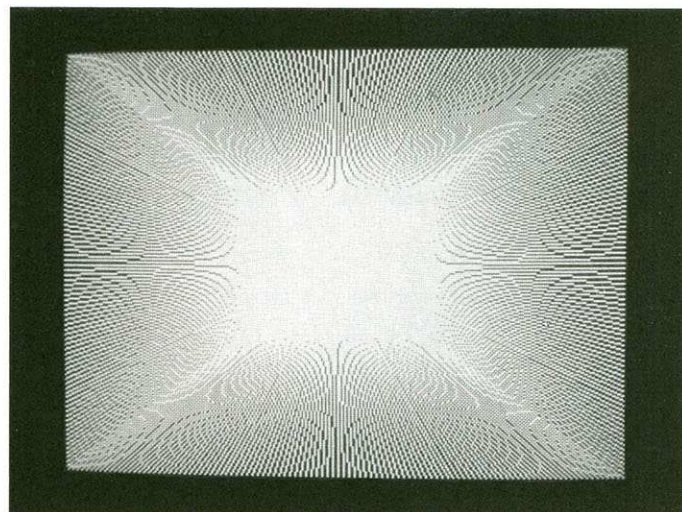


Bild 2. Das Ergebnis unseres Beispielprogramms

BLTAPTL: Hier den Wert von '2*KDelta-GDelta' eintragen.

BLTCPTx: enthält die Startadresse der Linie (Anfangswort). Die Berechnung dieses Anfangswortes erfolgt folgendermaßen:

Anfangsadresse der Bitplane + (Anzahl der Zeilen-Y1-1)* Bytes pro Zeile + 2*(X1/16)

Der nicht teilbare Rest der Division X1/16 wird normalerweise in die STARTx-Bits eingetragen:

REST = X1-INT(X1/16)*16

BLTDPTx: Muß ebenfalls (!) das Anfangswort der Linie enthalten (= BLTCPTx)

BLTAMOD: Hier den Wert von '2*KDelta-2*GDelta' eintragen.

BLTBMOD: Wert von '2*KDelta'.

BLTCMOD: Hier die Gesamtbreite der Bitplane in Bytes eintragen.

BLTDMOD: Den gleichen Wert wie bei BLTCMOD eintragen.

BLTADAT: muß fest mit $8000 initialisiert werden.

BLTBDAT: enthält die Maske der Linie.

BLTAFWM: muß fest mit $FFFF initialisiert werden.

BLTSIZE: Die Breite muß fest mit $2 initialisiert werden.

Die Höhe entspricht korrekt immer dem Wert von GDelta (enthält die Höhe der Linie in Punkten).

ACHTUNG: *Das Zeichnen der Linie wird durch Beschreiben dieses Registers automatisch gestartet. Deshalb dieses Register immer als letztes beschreiben.*

Das Beispiellisting zu diesem Kursteil erzeugt ein recht nettes Muster. Damit man überhaupt mitbekommt, daß der Blitter etwas zeichnet, wurden einige Warteschleifen in das Programm eingebaut, die aber zur vollen Entfaltung der Geschwindigkeit auch weggelassen werden können.

Damit wären wir wieder einmal am Ende der Blitter-Ecke angelangt. In der nächsten und vorerst letzten Folge erzählen wir noch etwas über das Thema "Scrolling mit dem Blitter". Bis dahin noch viel Spaß mit der AMIGA DOS. (jb)

Vergleiche	Oktant	Code	Vergleiche	Oktant	Code
Y1 <= Y2 X1 <= X2 DeltaX >= DeltaY	0	110	Y1 >= Y2 X1 >= X2 DeltaX >= DeltaY	4	101
Y1 <= Y2 X1 <= X2 DeltaX <= DeltaY	1	001	Y1 >= Y2 X1 >= X2 DeltaX <= DeltaY	5	010
Y1 <= Y2 X1 >= X2 DeltaX <= DeltaY	2	011	Y1 >= Y2 X1 <= X2 DeltaX <= DeltaY	6	000
Y1 <= Y2 X1 >= X2 DeltaX >= DeltaY	3	111	X1 >= Y2 X1 <= X2 DeltaX >= DeltaY	7	100

Abb. 1. Die Vergleiche zwischen Anfangs- und Endpunkt

BLTCON1 $DFF042

Bit-Nr.	Name	Funktion
15	Texture3	Wert fr die Verschiebung der Maske.
14	Texture2	Entspricht normalerweise den Start0-3
13	Texture1	Bits, damit die Maske pixelgetreu
12	Texture0	verluft.
11-7	No Name	immer auf Null setzen !
6	SIGN	Wenn 2*Kdelta<Gdelta ist, auf 1 setzen
5	No Name	immer auf Null setzen !
4	SUL	Diese drei Bits mssen mit den Code-
3	SUD	Bits der Oktanten initialisiert
2	AÚL	werden !
1	SING	Wenn nur ein Punkt pro Zeile, dann auf 1
0	LINE = 1	Damit der Linienmodus aktiv ist.

Abb. 2. Die Belegung des BLTCON1-Registers

BLTCON0 $DFF040

Bit-Nr.	Name	Funktion
15	START3	Diese 4 Bits enthalten die Position des Start-
14	START2	punktes innerhalb des Anfangswortes.
13	START1	
12	START0	
11	USEA=1	Die USEx-Bits mssen immer im Linienmodus diese
10	USEB=0	notwendige Kombination beinhalten.
9	USEC=1	
8	USED=1	
7-0	LF7-LF0	MuÃ mit $CA fest installiert werden (D = aC + AB)

Abb. 3. Die Belegung des BLTCON0-Registers

Listings

```
        opt l-

****************************************
* BLITTER-KURS fuer AMIGA-DOS Teil 5 *
*    by Ulf Petersen in 1990         *
*  Listing erstellt mit DEVPAC II    *
****************************************

```

Listing: Blitterkurs

DATABOX

```
BLTAPTL     equ     $52     Wert fuer 2*KDelta - GDelta
BLTCPTH     equ     $48     Anfangsadresse der Linie
BLTDPTH     equ     $54        "      "    "
BLTAMOD     equ     $64     Wert fuer 2*KDelta - 2*GDelta
BLTBMOD     equ     $62     Wert fuer 2*KDelta
BLTCMOD     equ     $60     Breite der Bitplane
BLTDMOD     equ     $66        "
BLTCON0     equ     $40     Kontrollregister 1
BLTCON1     equ     $42     Kontrollregister 2
BLTSIZE     equ     $58     Groesse des Blitterfensters
BLTAFWM     equ     $44     Fest mit $FFFF initialisiert
```

Listing: Blitterkurs

```
BLTADAT     equ    $74
BLTBDAT     equu   $72       Maske der Linie
COP1LC      equ    $80       Erste Copper-Liste
COP2LC      equ    $84       Zweite    "
COPJMP1     equ    $88       Initialisierungsregister Liste 1
COPJMP2     equ    $8a                 "               "    2
DIWSTRT     equ    $8e       Anfang des Bildschirmfensters
DIWSTOP     equ    $90       Ende des Bildschirmfensters
DDFSTRT     equ    $92       Bitplane-DMA   Anfang
DDFSTOP     equ    $94       Bitplane-DMA   Ende
BPL1PTH     equ    $e0       Adresse der ersten Bitplane
BPL1PTL     equ    $e2          "
BPL2PTH     equ    $e4       Adresse der zweiten Bitplane
BPL2PTL     equ    $e6          "
BPLCON0     equ    $100       Kontrollregister der Bitplane
BPLCON1     equ    $102       Scrolling
BPLCON2     equ    $104       Prioritaeten Grafik/Sprites
BPL1MOD     equ    $108       Modulo-Wert ungerader Bitplanes
BPL2MOD     equ    $10a        "      gerader Bitplanes
SPR0PTH     equ    $120       Adressen der Spritetabellen
SPR0PTL     equ    $122
CIAAPRA     equ    $BFE001
DMACON      equ    $96        DMA-Kontrollregister (write)
DMACONR     equ     2                   "            (read)
STARTLIST   equ    38
RANDOM      equ     6
; Library-Adressen
Forbid      equ    -30-102
Permit      equ    -30-108
AllocMem    equ    -30-168
FreeMem     equ    -30-180
OpenLibrary equ    -30-522
Exec        equ    4
ClearChip   equ    2+$10000
Size1       equ    40*256       Groesse der Bitplane
Size2       equ    100            "      "  Copper-Liste
; *** Start des Programms ***
START     move.l    Exec,a6
; Speicherplatz fuer die Bitplanes anfordern ...
     move.l    #Size1,d0
     move.l    #ClearChip,d1
     jsr       AllocMem(a6)
     move.l    d0,Bitplane
; Speicherplatz fuer die Copper-Liste anfordern ...
     move.l    #Size2,d0
     move.l    #ClearChip,d1
     jsr       AllocMem(a6)
     move.l    d0,Liste
; Copper-Liste direkt erstellen ...
     move.l    Liste,a0
     move.l    Bitplane,d0
; Bitplaneadresse in die Copper-Liste eintragen
     move.w    #BPL1PTL,(a0)+
     move.w    d0,(a0)+
     swap      d0
     move.w    #BPL1PTH,(a0)+
     move.w    d0,(a0)+
; Alle Sprites auf leere Datentabellen umlenken ...
     move.l    Liste,d0
     add.l     #96,d0
     move.l    #SPR0PTL,d1
     move.l    #SPR0PTH,d2
     moveq     #7,d3
Next      move.w    d1,(a0)+
     move.w    d0,(a0)+
     add.l     #4,d1
     swap      d0
     move.w    d2,(a0)+
     move.w    d0,(a0)+
     add.l     #4,d2
     swap      d0
     dbf       d3,Next
     move.l    #$fffffffe,(a0)       Ende der Copper-Liste
     lea       $dff000,a5
     move.l    Liste,COP1LC(a5)      Aktivierung der neuen
     clr.w     COPJMP1(a5)           Copper-Liste
     move.l    $6C,OldInt            Interrupt-Vektor
     move.l    #NewInt,$6c           verbiegen
     move.w    #$3081,DIWSTRT(a5)    Standardbildschirm auf-
     move.w    #$30c1,DIWSTOP(a5)    bauen mit einer
     move.w    #$0038,DDFSTRT(a5)    Bitplane.
     move.w    #$00d0,DDFSTOP(a5)
     move.w    #%0001001000000000,BPLCON0(a5)
```

Listing: Blitterkurs

```
     clr.w     BPLCON1(a5)
     clr.w     BPLCON2(a5)
     move.w    #$83e0,DMACON(a5)
; Neue Hintergrundfarben
     clr.w     $dff180
     move.w    #$46e,$dff182
; Hauptprogramm: Die am Anfang stehende Warteschleife kann
; ausgelassen werden, um den Blitter in Echtzeit arbeiten
; zu lassen. Loeschen Sie dazu bloss
; "bne.s Go_on" und "bne.s Go_on2",
; sowie "bne.s Go_on3" und "bne.s Go_on4".
     move.w    #0,X1                 Anfangs- und Endpunkt
     move.w    #0,Y1
     move.w    #320,X2               der Linie festlegen
     move.w    #256,Y2
Go_on      cmp.b    #0,FLAG
           bne.s    Go_on
Go_on2     cmp.b    #0,FLAG
           beq.s    Go_on2
     move.w    X1,d0                 Register initialisieren
     move.w    Y1,d1
     move.w    X2,d2
     move.w    Y2,d3
     jsr       ZEICHNE               und Linie zeichnen
     addq      #3,X1                 Werte um 3 erhqehen, um
     subq      #3,X2                 ein Muster zu erzeugen
     cmp.w     #321,X1               Fertig ?
     bne.s     Go_on
     move.w    #0,X1                 Genau das gleiche noch
     move.w    #0,Y1                 einmal fuer die Seiten-
     move.w    #320,X2               flaechen
     move.w    #256,Y2
Go_on3     cmp.b    #0,FLAG
           bne.s    Go_on3
Go_on4     cmp.b    #0,FLAG
           beq.s    Go_on4
     move.w    X1,d0
     move.w    Y1,d1
     move.w    X2,d2
     move.w    Y2,d3
     jsr       ZEICHNE
     addq      #3,Y1
     subq      #3,Y2
     cmp.w     #258,Y1
     bne.s     Go_on3
LAUFE      btst     #6,CIAAPRA       Linke Maustaste gedr.?
     bne.s     LAUFE                 Nein, also warten !
     jmp       EXIT
; PARAMETER fuer die Linie ...
; d0 = X1    d1 = Y1                 a0 = Anfangsadresse der
; d2 = X2    d3 = Y2                      Bitplane
; Anfangsadresse der Linie berechnen
ZEICHNE    clr.l    d4
     clr.l     d5
     clr.l     d6
     move.l    Bitplane,a0
     move.l    #40,d4
     mulu      d1,d4                 Y1 * Breite der Bitplane
     moveq     #-$10,d5
     and.w     d0,d5                 Bits 0-3 ausmaskieren
     lsr.w     #3,d5                 dann durch 8 teilen
     add.w     d5,d4
     add.l     a0,d4                 Fertige Anfangsadresse
     clr.l     d5                    d5 freimachen
     sub.w     d1,d3                 DeltaY nach d3
     roxl.b    #1,d5                 Vorzeichen nach d5
     tst.w     d3
     bge.s     Positiv1              d3 bereits positiv
     neg.w     d3                    Sonst DeltaY positiv
                                     ;machen
Positiv1   sub.w    d0,d2            DeltaX nach d2
     roxl.b    #1,d5                 Vorzeichen nach d5
     tst.w     d2
     bge.s     Positiv2              d2 bereits positiv
     neg.w     d2                    Sonst DeltaX positiv
                                     ;machen
Positiv2   move.w   d3,d1            DeltaY nach d1
     sub.w     d2,d1                 DeltaY - DeltaX
     bge.s     Pos1                  Wenn DeltaY > DeltaX : w
eiter
     exg       d2,d3                 Groesseres Delta (GDelta
) nach d3
Pos1       roxl.b   #1,d5
     lea       Tabelle,a1
     add.l     d5,a1
     move.b    (a1),d5
     mulu      #2,d2                 KDelta * 2
     move.w    d2,BLTBMOD(a5)
     sub.w     d3,d2                 2 * KDelta - GDelta
```



Listing: Blitterkurs

```
        bge.s     Pos2
        or.b      #64,d5              SIGN-Bit setzen
Pos2        move.w  d2,BLTAPTL(a5)
        sub.w     d3,d2               2 * KDelta - 2 * GDelta
        move.w    d2,BLTAMOD(a5)

        move.l    d4,BLTCPTH(a5)      Anfangsadresse
        move.l    d4,BLTDPTH(a5)
        move.w    #40,BLTCMOD(a5)     Breite der Bitplane
        move.w    #40,BLTDMOD(a5)
        move.w    #$ffff,BLTAFWM(a5)
        move.w    #$ffff,BLTBDAT(a5)  Maske der Linie
        move.w    #$8000,BLTADAT(a5)
        and.w     #$000f,d0
        ror.w     #4,d0
        or.w      #$0bca,d0
        move.w    d0,BLTCON0(a5)
        move.w    d5,BLTCON1(a5)

; BLTSIZE berechnen

        lsl.w     #6,d3               GDelta mal 64 = Laenge
        addq.w    #2,d3
        move.w    d3,BLTSIZE(a5)

wait        btst    #14,DMACONR(a5) Warteschleife, bis der
        bne.s     wait                Blitter fertig ist.

        rts


; Die am Anfang reservierten Speicherbereiche wieder fuer
; eine weitere Verwendung freigeben.

EXIT        move.l  #Size2,d0         Copper-Liste
        move.l    Liste,a1
        jsr       FreeMem(a6)

        move.l    #Size1,d0           Bitplane
        move.l    Bitplane,a1
        jsr       FreeMem(a6)

; Alte Copper-Liste wieder aktivieren ...

        lea       GRFX,a1
        clr.l     d0
        jsr       OpenLibrary(a6)
        move.l    d0,a4
        move.l    STARTLIST(a4),COP1LC(a5)
```



Listing: Blitterkurs

```
        clr.w     COPJMP1(a5)

        move.l    OldInt,$6C              Alter IRQ-Vektor

        move.w    #$83e0,DMACON(a5)

        clr.l     d0                      Alles ROGER !

        rts                               und raus ...

; CHIP-RAM Adressen

Bitplane  dc.l    0                       BITPLANE
Liste     dc.l    0                       COPPER-LISTE

GRFX      dc.b    "graphics.library",0

        cnop    0,2

FLAG      dc.b    0

        cnop    0,2

X1        dc.w    0                       Koordinaten vom Anfangspunkt
Y1        dc.w    0
X2        dc.w    0                       Koordinaten vom Endpunkt
Y2        dc.w    0

; OKTANTENTABELLE (zusaetzlich ist das LINE-Bit gesetzt)
Tabelle   dc.b    0*4+1,4*4+1,2*4+1,5*4+1,1*4+1,6*4+1,3*4+1,7*4+1
;                 OKT6  OKT7  OKT5  OKT4  OKT1  OKT0  OKT2  OKT3

; Der nun folgende Interrupt hat lediglich die Aufgabe, eine Zeitverzoegerung
; herbeizufuehren, damit man ueberhaupt etwas vom Zeichnen mitbekommt und damit
; der Rasterstrahl nicht mit dem Bildaufbau kollidiert.

NewInt    movem.l d0-d7/a0-a6,-(sp)
        add.b     #1,FLAG
        cmp.b     #3,FLAG
        bne.s     GOTO
        move.b    #0,FLAG
GOTO      movem.l (sp)+,d0-d7/a0-a6
          dc.w    $4ef9                   Code fuer JMP-Befehl
OldInt    dc.l    0                       Alter Vektor
```

Listing: Blitterkurs



Torsten Budesheim

External-Boot – Systemstart außer Haus

Horch, was kommt von draußen rein...

Der Bootvorgang von Diskette war bisher immer nur von dem internen, beziehungsweise dem als Nummer eins angemeldeten Laufwerk möglich. Selbst unter der Kickstartversion 1.3 ist es sonst höchstens noch von einer über SCSI-Schnittstelle laufenden Festplatte möglich. Hier soll nun mit dem vorliegenden Programm 'External-Boot' eine softwaremäßige Abhilfe geschaffen werden.

Wie das funktioniert, erklären wir Ihnen gleich, nur eins vorweg: es ist nicht bei allen Programmen möglich! Wie jeder weiß, der schon ein wenig mit dem Amiga programmiert hat, werden Systemaufrufe über sogenannte Einsprungtabellen getätigt. Dies ermöglicht es, das Betriebssystem (Kickstart) zu modifizieren, ohne daß Programme, die unter der alten Kickstartversion erstellt wurden, abstürzen. So stehen in der Execbase-Struktur zwei Sprungmarken, über die auch vom Betriebssystem beim Booten verzweigt wird. Es sind dies die Vektoren für die Betriebssystemroutinen 'OpenResource' und 'OpenDevice'. Das Betriebssystem öffnet nämlich das 'trackdisk.device' für das als Null angemeldete Laufwerk (im Normalfall das interne df0:), um dann von diesem die Boot-Blöcke zu laden und in ein eventuell vorhandenes Bootprogramm zu verzweigen. Genau dies machen wir uns einfach zunutze, um von externen Laufwerken zu booten. Wir vergleichen einfach, ob es sich um einen Zugriff auf das 'trackdisk.device' handelt, und wenn ja, so wird eventuell die Unit 0 gegen unsere Unitnummer ausgetauscht. Nun werden Sie sich vielleicht fragen, was das 'Umbiegen' der 'OpenResource'-Routine dabei noch soll? Ganz einfach, es gibt Programme (unter anderem auch das Betriebssystem), die für kurze Zeit ein Laufwerk für andere Tasks mittels der 'AllocUnit'-Funktion der 'disk.resource'-Struktur unzugänglich machen. Damit auch wirklich das richtige Laufwerk 'belegt' wird, müssen folglich noch Vektoren 'umgebogen' werden. Sie können sich jetzt wahrscheinlich auch schon denken, warum dies nicht bei allen Programmen funktioniert. Genau, weil nämlich manche Programme andere Tasks abschalten und dann direkt auf die Hardware zugreifen, um Daten zu laden und somit erst gar nicht über das Betriebssystem springen.

Warum das alles, wird sich jetzt wohl so mancher fragen. Nun, einige Amiga-User besitzen sicherlich als externes Laufwerk ein 5,25"-Laufwerk, und da diese Disketten immer noch um einiges billiger sind als die 3,5-"Zöller", ist es doch von Vorteil, so viel Programme wie möglich auf das externe 5,25"-Laufwerk zu kopieren.

Warum von externen Laufwerken booten?

Dies ist jetzt also auch mit (vielen) selbstbootenden Programmen möglich. Nun aber ein wenig zum Umgang mit dem Programm:

Wenn Sie das Programm starten, öffnet sich ein Fenster, in dem Sie festlegen können, von welchem Laufwerk Sie booten möchten, welche Funktionen Sie 'umbiegen' wollen, und welche Funktion nach einmaligem Aufruf wieder normal arbeiten soll.

Zum letzten Punkt folgende Erläuterung: Wenn Sie hier 'beide' einstellen, so lädt der Amiga nur die Boot-Blöcke von einem externen Laufwerk und fährt dann ganz normal mit seiner Arbeit (wie gewohnt) fort. So ist es hiermit zum Beispiel leicht möglich, Boot-Block-Viren zu umgehen: indem Sie nämlich in das externe Laufwerk eine schreibgeschützte Diskette legen (mit einem nicht infizierten BootBlock) und in das interne Laufwerk Ihre nicht schreibgeschützte Arbeitsdiskette, auf die sich ja bei Betrieb eventuell ein Virus schreiben könnte!

Sollten Sie das Programm einmal verlassen wollen, ohne daß es sich im System installiert, so können Sie das 'Cancel'- oder das 'Close'-Gadget betätigen. Das Programm arbeitet über den 'CoolCapture'-Vektor der Execbase-Struktur und bleibt nur für einen Resetvorgang im System. Dies wurde mit Absicht so gemacht, um nicht mit Virus-Checkern im Boot-Block oder der Startup-Sequence in Konflikt zu kommen.

Das Programm wurde mit dem Devpac-Assembler 2.0 erstellt, wobei auf größere Include-Einbindungen verzichtet wurde, um nicht immer beim Assemblieren auf eine eventuell vorhandene Festplatte warten zu müssen. So, nun aber viel Spaß beim Booten von den Externen, wünscht Ihnen das AMIGA DOS-Team. (br)

Listings






```
*************************************************************
***                  ExternalBoot V1.0                    ***
***            written by Torsten Budesheim               ***
***               (c) 1990 by DMV-Verlag                  ***
***            Sprache: Assembler (Devpac)                ***
*************************************************************

* Exec-Offsets
ExecBase      = 4
OpenLibrary   = -30-522
CloseLibrary  = -30-384
OpenDevice    = -30-414
AllocMem      = -30-168
Wait          = -30-288
GetMsg        = -30-342
ReplyMsg      = -30-348
OpenResource  = -30-468

* Intuition-Offsets
OpenWindow      = -30-174
CloseWindow     = -30-42
RefreshGadgets  = -30-192

* Graphics-Offsets
Text          = -30-30
TextLength    = -30-24
Move          = -30-210
SetSoftStyle  = -30-60
SetAPen       = -30-312
SetFont       = -30-36
OpenFont      = -30-42
CloseFont     = -30-48
Draw          = -30-216

* Labels-Definitionen
GetUnitID     = -30
AllocUnit     = -6
FreeUnit      = -12
INTUI_REV     = 31
GRAF_REV      = 31
MEMF_PUBLIC   = 1
CoolCapture   = 46
UserPort      = 86
mp_SigBit     = 15
Class         = 20
IAddress      = 28
wd_RPort      = 50


   SECTION  "Programm",CODE
* Hauptprogramm
start:  move.l  ExecBase,a6
        move.l  #GRAF_REV,d0           ;Version
        lea     graf_name,a1
        jsr     OpenLibrary(a6)
        move.l  d0,GfxBase
        beq     exit

        move.l  #INTUI_REV,d0          ;Version
        lea     intui_name,a1
        jsr     OpenLibrary(a6)
        move.l  d0,IntuitionBase
        beq     close_gfx

        move.l  d0,a6
        lea     NewWindow,a0
        jsr     OpenWindow(a6)
        move.l  d0,window
        beq     close_intui

        move.l  d0,a0
        move.l  wd_RPort(a0),RPort     ;RastPort merken

        move.l  GfxBase,a6
        lea     Font,a0
        jsr     OpenFont(a6)
        move.l  d0,FontPtr
        beq     close_win
        move.l  d0,a0
        move.l  RPort,a1
        jsr     SetFont(a6)

        move.l  RPort,a1
        move.l  GfxBase,a6
        moveq   #1,d0
        jsr     SetAPen(a6)
        lea     Header1,a0
        move.b  #2,d0                  ;Font -> BOLD
        bsr     print
        lea     Header2,a0
        bsr     print
        lea     Header3,a0
        bsr     print

        move.l  RPort,a1
        moveq   #2,d0
        jsr     SetAPen(a6)
        move.l  RPort,a1
        move.l  #168,d1
        moveq   #13,d0
        jsr     Move(a6)
        move.l  RPort,a1
        move.l  #357,d0
        move.l  #168,d1
        jsr     Draw(a6)

        move.l  GfxBase,a6
        move.l  RPort,a1
        moveq   #3,d0
        jsr     SetAPen(a6)
        lea     Frage1,a0
        move.b  #1,d0
        bsr     print
        lea     Frage0,a0
        bsr     print
        lea     Frage2,a0
        bsr     print
        lea     Frage2_2,a0
        bsr     print
        bra.s   n2_2

warte   bsr     WhileMsg
        cmpi.l  #$200,d0                        ;CLOSEWINDOW?
        beq     close_font
        cmpi.l  #$40,d0
        bne.s   warte
        cmp.l   #CancelGadget,d1
        beq     close_font
        cmp.l   #Gadget1,d1
        bne.s   n1
        lea     beide_txt_info,a0
        move.l  #alloc_txt,d1
        move.l  #beide_txt,d0
        bsr     NewText
        bra.s   warte
n1      cmp.l   #Gadget2,d1
        bne.s   n2
        lea     keine_txt_info,a0
        move.l  #keine_txt,d1
        move.l  #beide_txt,d0
        bsr     NewText
        bra.s   warte
n2      cmp.l   #DriveGadget,d1
        bne.s   n3
n2_2    lea     drive_txt_info,a0
        move.l  #drive0_txt,d0
        move.l  #drive3_txt,d1
        bsr     NewText
        move.l  txt_ptr3-drive_txt_info(a0),a0
        moveq   #0,d0
        move.b  5(a0),d0
        subi.b  #48,d0
        move.l  d0,drive_number
        bsr     TestDrive
        tst.l   d0
        beq.s   n2_2
        bra     warte
n3      cmp.l   #OkGadget,d1
        bne.s   n4
        bsr     InstallReset
        bra.s   close_font
n4      bsr     InstallReset
        move.w  #$4000,$dff09a          ;Reset veranlassen
        move.l  #reset,$b4.w
        trap    #$d
reset   move.w  #$2700,sr
        jmp     $fc0000

close_font
        move.l  GfxBase,a6
        move.l  FontPtr,a1
        jsr     CloseFont(a6)
close_win
        move.l  IntuitionBase,a6
        move.l  window,a0
        jsr     CloseWindow(a6)
close_intui
        move.l  ExecBase,a6
        move.l  IntuitionBase,a1
        jsr     CloseLibrary(a6)
close_gfx
        move.l  ExecBase,a6
        move.l  GfxBase,a1
        jsr     CloseLibrary(a6)
exit    moveq   #0,d0
        rts

* Schreibt neuen Text in Gadget
* A0: Zeiger auf txt_info
* D0: Zeiger auf ersten Text
* D1: Zeiger auf letzten Text
NewText
        movem.l  d0/d1/a0-a2,-(sp)

        move.l  txt_ptr3-drive_txt_info(a0),a1
        cmp.l   d1,a1
        bne.s   not_same
        move.l  d0,txt_ptr3-drive_txt_info(a0)
        bra.s   same
not_same
        lea     11(a1),a1
```

Listing: ExternalBoot.s




```
        move.l  a1,txt_ptr3-drive_txt_info(a0)
same    clr.b   FontStyle
        move.l  IntuitionBase,a6
        move.l  window,a1
        sub.l   a2,a2
        lea     Gadget1,a0
        jsr     RefreshGadgets(a6)

        movem.l (sp)+,d0/d1/a0-a2
        rts

* Testet, ob Laufwerk vorhanden ist
* D0: Unit
* Rückgabe;
* D0: NULL bei Fehler
TestDrive
        move.l  d0,-(sp)
        move.l  ExecBase,a6
        lea     resource_name,a1
        moveq   #0,d0
        jsr     OpenResource(a6)
        tst.l   d0
        beq.s   end_test
        move.l  d0,a6
        move.l  (sp),d0
        jsr     GetUnitID(a6)
        cmp.l   #$ffffffff,d0       ;Laufwerk vorhanden?
        beq.s   no_drive
        moveq   #0,d0
no_drive
        addq.l  #1,d0
end_test
        move.l  (sp)+,d1
        rts

* Druckt ein String in RastPort
* D0: FontStyle
* A0: Zeiger auf Text
print   movem.l d0-d2/a0/a1,-(sp)

        bsr.s   WindowsFontStyle    ;Font-Style setzen
        moveq   #0,d1
        move.w  (a0)+,d1
        bsr.s   strlen              ;Stringlänge holen
        move.l  RPort,a1
        movem.l d0/d1/a0/a1,-(sp)
        move.l  GfxBase,a6
        jsr     TextLength(a6)
        moveq   #0,d2
        move.w  wd_width,d2
        sub.l   d0,d2
        divu    #2,d2
        movem.l (sp),d0/d1/a0/a1
        move.l  d2,d0
        jsr     Move(a6)
        movem.l (sp)+,d0/d1/a0/a1
        jsr     Text(a6)

        movem.l (sp)+,d0-d2/a0/a1
        rts

* Ermittelt die Länge des Strings
* A0: Zeiger auf String
* Rückgabe:
* D0: Länge des Strings
strlen  move.l  a0,-(sp)

        moveq   #-1,d0
getlen  addq.l  #1,d0
        tst.b   (a0)+
        bne.s   getlen

        move.l  (sp)+,a0
        rts

* Verändert den FontStyle für das Window
* D0: Style
WindowsFontStyle
        movem.l d0/a0/a1,-(sp)

        and.l   #$ff,d0             ;Neuer Fontstyle
        move.l  RPort,a1
        moveq   #-1,d1              ;Enable
        move.l  GfxBase,a6
        jsr     SetSoftStyle(a6)

        movem.l (sp)+,d0/a0/a1
        rts

* Wartet auf Aktivitäten im eigenen Window
* Rückgabe:
* D0: Class
* D1: IAddress
WhileMsg
        movem.l d6/d7/a0/a1/a6,-(sp)

        move.l  window,a0
        move.l  UserPort(a0),a0
        move.b  mp_SigBit(a0),d1
        moveq   #1,d0
        lsl.l   d1,d0
        move.l  ExecBase,a6
```

Listing: ExternalBoot.s

```
        jsr     Wait(a6)

MsgLoop
        move.l  window,a0
        move.l  UserPort(a0),a0
        jsr     GetMsg(a6)
        tst.l   d0
        beq.s   WhileMsg

        move.l  d0,a1
        move.l  Class(a1),d7
        move.l  IAddress(a1),d6
        jsr  ReplyMsg(a6)

        move.l  d7,d0
        move.l  d6,d1
        movem.l (sp)+,d6/d7/a0/a1/a6
        rts

* Installiert Reset-Routine
InstallReset
        move.l  ExecBase,a6
        cmp.l   #beide_txt,txt_ptr1
        bne.s   ncmp1
        move.b  #1,device_mod
        move.b  #1,alloc_mod
        bra.s   inst1_end
ncmp1   cmp.l   #alloc_txt,txt_ptr1
        bne.s   ncmp2
        move.b  #1,alloc_mod
        clr.b   device_mod
        bra.s   inst1_end
ncmp2   clr.b   alloc_mod
        move.b  #1,device_mod
inst1_end
        cmp.l   #beide_txt,txt_ptr2
        bne.s   ncmp3
        move.b  #1,alloc_clear
        move.b  #1,device_clear
        bra.s   inst2_end
ncmp3   cmp.l   #keine_txt,txt_ptr2
        bne.s   ncmp4
        clr.b   alloc_clear
        clr.b   device_clear
        bra.s   inst2_end
ncmp4   cmp.l   #alloc_txt,txt_ptr2
        bne.s   ncmp5
        move.b  #1,alloc_clear
        clr.b   device_clear
        bra.s   inst2_end
ncmp5   clr.b   alloc_clear
        move.b  #1,device_clear
inst2_end
        jsr     inst_reset
        rts

   SECTION  "Reset-Routine",DATA
* Hier beginnt das Programm, das in die
* Reset-Routine eingebunden wird
NewReset:
        bsr.s   inst_reset
        tst.l   d0
        bne.s   no_reset_error
        rts
no_reset_error
        clr.l   CoolCapture(a6)
        addi.l  #go_on_reset-NewReset,d0
        move.l  d0,a0
        jmp     (a0)
go_on_reset
        movem.l d0-d7/a0-a6,-(sp)
        move.b  device_mod(pc),d1
        tst.b   d1
        beq.s   no_device_modify
        lea     NewOpenDevice(pc),a1
        lea     jmp_OD+2(pc),a0
        move.l  OpenDevice+2(a6),(a0)    ;alten Zeiger
 retten
        move.l  a1,OpenDevice+2(a6)
no_device_modify
        move.b  alloc_mod(pc),d1
        tst.b   d1
        beq.s   no_alloc_modify
        lea     NewOpenResource(pc),a2
        lea     jmp_OR+2(pc),a0
        lea     jsr_OR+2(pc),a1
        move.l  OpenResource+2(a6),(a0) ;alten Zeiger
 retten
        move.l  OpenResource+2(a6),(a1)
        move.l  a2,OpenResource+2(a6)
no_alloc_modify
no_mem
        movem.l (sp)+,d0-d7/a0-a6
        rts

inst_reset
        movem.l d1-d7/a0-a6,-(sp)

        move.l  #EndReset-NewReset+1,d0 ;Mem-Size für
 'AllocMem'
        moveq   #MEMF_PUBLIC,d1  ;Speicherbereich egal
        jsr     AllocMem(a6)
        move.l  d0,CoolCapture(a6)   ;Zeiger speichern
        beq.s   end_inst
```

Listing: ExternalBoot.s

```
        move.l  d0,-(sp)

        move.l  d0,a1                   ;Destination
        move.l  #EndReset-NewReset,d0   ;Size
        lea     NewReset(pc),a0         ;Source
l1      move.b  (a0)+,(a1)+
        dbf     d0,l1

        moveq   #0,d0
        lea     34(a6),a0
        move.w  #$16,d1
l2      add.w   (a0)+,d0
        dbf     d1,l2
        not.w   d0
        move.w  d0,82(a6)

        move.l  (sp)+,d0
end_inst
        movem.l (sp)+,d1-d7/a0-a6
        rts

* Hier kommt die modifizierte 'OpenDevice'-Routine
NewOpenDevice
        movem.l d1/d2/a0/a1,-(sp)
        lea     device_name(pc),a1
cmp_td
        move.b  (a1)+,d2    ;Byte aus trackdisk.device
        move.b  (a0)+,d1    ;Byte aus Device-Name (ROM)
        cmp.b   d2,d1           ;identisch?
        bne.s   no_dev_clear    ;Nein? Dann abbrechen
        tst.b   (a0)            ;Ende des Device-Namens?
        bne.s   cmp_td  ;Nein? Dann weiter vergleichen
        bsr.s   changeUnit
        move.b  device_clear(pc),d1
        tst.b   d1                  ;muß Vektor wieder
        beq.s   no_dev_clear        ;hergestellt werden?
        move.l  jmp_OD+2(pc),OpenDevice+2(a6);alten Vektor herstellen
no_dev_clear
        movem.l (sp)+,d1/d2/a0/a1
jmp_OD  jmp     $ffffffe

device_name
        dc.b    'trackdisk.device',0
resource_name
        dc.b    'disk.resource',0
device_clear
        dc.b    0
alloc_clear
        dc.b    0
device_mod
        dc.b    1
alloc_mod
        dc.b    1
  EVEN
drive_number
        dc.l    0

* Ersetzt 'alte' Unit-Nr. durch die eigene
changeUnit
        tst.l   d0                      ;Unit-Nr. 0?
        bne.s   no_intern_boot          ;Nein? Dann weiter
        move.l  drive_number(pc),d0 ;eigene Drive-Nr. holen
        bra.s   end_changeUnit
no_intern_boot
        move.l  d1,-(sp)
        move.l  drive_number(pc),d1
        cmp.l   d1,d0                   ;Gleiche Unit-Nr.?
        bne.s   not_same_Unit ;Nein? Dann überspringen
        moveq   #0,d0                   ;ansonsten -> Unit 0
not_same_Unit
        move.l  (sp)+,d1
end_changeUnit
        rts

* Hier beginnt die modifizierte 'AllocUnit'-Routine
NewAllocUnit
        bsr.s   changeUnit
        move.l  d1,-(sp)
        move.b  alloc_clear(pc),d1     ;nach einmaligem
        tst.b   d1                     ;Aufruf zurücksetzen?
        beq.s   no_alloc_clear;Nein? Dann überspringen
        move.l  jmp_AU+2(pc),AllocUnit+2(a6)
no_alloc_clear
        move.l  (sp)+,d1
jmp_AU  jmp  $ffffffe

* Hier beginnt die modifizierte 'FreeUnit'-Routine
NewFreeUnit
        bsr.s   changeUnit
        move.l  d1,-(sp)
        move.b  alloc_clear(pc),d1     ;nach einmaligem
        tst.b   d1                     ;Aufruf zurücksetzen?
        beq.s   no_free_clear ;Nein? Dann überspringen
        move.l  jmp_FU+2(pc),FreeUnit+2(a6)
no_free_clear
        move.l  (sp)+,d1
jmp_FU  jmp     $ffffffe

* Hier beginnt die modifizierte 'GetUnitID'-Routine
NewGetUnitID
        bsr.s   changeUnit
        move.l  d1,-(sp)
        move.b  alloc_clear(pc),d1     ;nach einmaligem
        tst.b   d1                     ;Aufruf zurücksetzen?
```



Listing: ExternalBoot.s

Nürnberger Public Domain Studio
Humboldtstr.141 im CCC
8500 Nürnberg 40
Tel.:0911/45 77 54 Fax : 0911 446 72 79

Ca.**9000**
Public Domain + Shareware
Disketten für
Amiga , Atari ST und PC.
Gratis Katalog - Disk und Info
für Ihren Computer anfordern !

Farbige Disketten

5.25" 2DD

	10 Stück	100 Stück
Schwarz	DM5.90	DM54.90
Weiss	DM9.90	DM94.90
Rot	DM9.90	DM94.90
Grau	DM9.90	DM94.90
Gelb	DM9.90	DM94.90
Blau	DM9.90	DM94.90
Orange	DM9.90	DM94.90

3.5" 2DD

	10 Stück	50 Stück
Blau	DM11.90	------------
Grau	DM11.90	------------
Rot	DM17.90	DM85.00
Gelb	DM17.90	DM85.90
Grün	DM17.90	DM85.90

5.25" HD
10 Stück 5.25" HD DM 12.80

LEERDISKETTEN SIND NUR ÜBER VERSAND ZU BEZIEHEN

Versandkosten:
Nachnahme DM 6.- Vorkasse DM3.-

```
        beq.s   no_get_clear   ;Nein? Dann überspringen
        move.l  jmp_GU+2(pc),GetUnitID+2(a6)
no_get_clear
        move.l  (sp)+,d1
jmp_GU  jmp     $ffffffe

* Hier beginnt die modifizierte 'OpenResource'-Routine
NewOpenResource
        movem.l d1/d2/a0/a1,-(sp)
        lea     resource_name(pc),a0
cmp_dr
        move.b  (a1)+,d2        ;Byte aus disk.resource
        move.b  (a0)+,d1 ;Byte aus Resource-Name (ROM)
        cmp.b   d2,d1           ;identisch?
        bne.s   no_res_clear   ;Nein? Dann abbrechen
        tst.b   (a1)            ;Ende des Device-Namens?
        bne.s   cmp_dr   ;Nein? Dann weiter vergleichen
        movem.l (sp)+,d1/d2/a0/a1
jsr_OR  jsr     $ffffffe
        tst.l   d0
        beq.s   no_open_res
        movem.l d1/a0-a2,-(sp)
        move.l  d0,a1           ;Zeiger auf Resource-Base

        lea     jmp_AU+2(pc),a2
        lea     NewAllocUnit(pc),a0
        move.l  AllocUnit+2(a1),(a2) ;Zeiger auf 'AllocUnit' sichern
        move.l  a0,AllocUnit+2(a1)   ;Neuen Zeiger speichern

        lea     jmp_FU+2(pc),a2
        lea     NewFreeUnit(pc),a0
        move.l  FreeUnit+2(a1),(a2)  ;Zeiger auf 'FreeUnit' sichern
        move.l  a0,FreeUnit+2(a1)    ;Neuen Zeiger speichern

        lea     jmp_GU+2(pc),a2
        lea     NewGetUnitID(pc),a0
        move.l  GetUnitID+2(a1),(a2)
        move.l  a0,GetUnitID+2(a1)

        move.b  alloc_clear(pc),d1 ;Muß Routine wieder
        tst.b   d1              ;'hingebogen' werden?
        beq.s   no_OR_clear     ;Nein? Dann überspringen
        move.l  jmp_OR+2(pc),OpenResource+2(a6)
no_OR_clear
        movem.l (sp)+,d1/a0-a2
no_open_res
        rts
no_res_clear
        movem.l (sp)+,d1/d2/a0/a1
jmp_OR  jmp     $ffffffe

EndReset

   SECTION  "Daten",DATA
koordinaten1
        dc.w    -1,-1,82,-1,82,10,-1,10,-1,-1

border1
        dc.w    0,0
        dc.b    2,1,0,5
        dc.l    koordinaten1,0

beide_txt_info
        dc.b    1,0,1,0
        dc.w    1,1
        dc.l    Font
txt_ptr1
        dc.l    beide_txt,0
keine_txt_info
        dc.b    1,0,1,0
        dc.w    1,1
        dc.l    Font
txt_ptr2
        dc.l    keine_txt,0
cancel_txt_info
        dc.b    1,0,1,0
        dc.w    17,1
        dc.l    Font,cancel_txt,0
ok_txt_info
        dc.b    1,0,1,0
        dc.w    34,1
        dc.l    Font,ok_txt,0
reset_txt_info
        dc.b    1,0,1,0
        dc.w    21,1
        dc.l    Font,reset_txt,0
drive_txt_info
        dc.b    1,0,1,0
        dc.w    1,1
        dc.l    Font
txt_ptr3
        dc.l    drive2_txt,0

DriveGadget
        dc.l    0
        dc.w    144,65,82,10
        dc.w    1,3,1
        dc.l    border1,0,drive_txt_info,0,0
        dc.w    6,0,0
CancelGadget
        dc.l    DriveGadget
        dc.w    268,186,82,10
```

Listing: ExternalBoot.s

```
        dc.w    0,3,1
        dc.l    border1,0,cancel_txt_info,0,0
        dc.w    5,0,0
OkGadget
        dc.l    CancelGadget
        dc.w    20,194,82,10
        dc.w    0,3,1
        dc.l    border1,0,ok_txt_info,0,0
        dc.w    4,0,0
ResetGadget
        dc.l    OkGadget
        dc.w    20,178,82,10
        dc.w    0,3,1
        dc.l    border1,0,reset_txt_info,0,0
        dc.w    3,0,0
Gadget2
        dc.l    ResetGadget
        dc.w    144,148,82,10
        dc.w    1,3,1
        dc.l    border1,0,keine_txt_info,0,0
        dc.w    2,0,0
Gadget1
        dc.l    Gadget2
        dc.w    144,102,82,10
        dc.w    1,3,1
        dc.l    border1,0,beide_txt_info,0,0
        dc.w    1,0,0

NewWindow
        dc.w    150,21
wd_width
        dc.w    370,213
        dc.b    -1,-1
        dc.l    $240
        dc.l    $3100e
        dc.l    Gadget1
        dc.l    0,NewWindowName
        dc.l    0,0,0,0
        dc.w    1
NewWindowName
        dc.b    'ExternalBoot V1.0',0
intui_name
        dc.b    'intuition.library',0
FontName
        dc.b    'topaz.font',0
graf_name
        dc.b    'graphics.library',0
cancel_txt
        dc.b    'Cancel',0
reset_txt
        dc.b    'Reset',0
ok_txt
        dc.b    'Ok',0
beide_txt
        dc.b    '  Beide   ',0
open_txt
        dc.b    'OpenDevice',0
alloc_txt
        dc.b    ' OpenRes  ',0
keine_txt
        dc.b    '  Keine   ',0
drive0_txt
        dc.b    '   DF0:   ',0
drive1_txt
        dc.b    '   DF1:   ',0
drive2_txt
        dc.b    '   DF2:   ',0
drive3_txt
        dc.b    '   DF3:   ',0
   EVEN
Header1 dc.b    0,18,'Booten von externen'
        dc.b    ' Laufwerken!',0
   EVEN
Header2 dc.b    0,28,'- Durch modifizieren von -',0
   EVEN
Header3 dc.b    0,37,'- ',39,'OpenDevice',39
        dc.b    ' und/oder ',39,'OpenResource',39
        dc.b    ' -',0
   EVEN
Frage0  dc.b    0,55,'Von welchem Laufwerk soll'
        dc.b    ' gebootet werden?',0
   EVEN
Frage1  dc.b    0,93,'Welche Routine soll modifiziert'
        dc.b    ' werden?',0
   EVEN
Frage2  dc.b    0,131,'Welche Routine soll nach'
        dc.b    ' einmaligem',0
   EVEN
Frage2_2
        dc.b    0,140,'Aufruf wieder unmodifiziert'
        dc.b    ' arbeiten?',0
   EVEN
Font    dc.l    FontName
        dc.w    8
FontStyle
        dc.b    0,1

   SECTION  "Reserve",BSS
IntuitionBase
        ds.l    1
GfxBase ds.l    1
FontPtr ds.l    1
RPort   ds.l    1
window  ds.l    1

   END
```

Listing: ExternalBoot.s

Edgar Meyzies

Tips und Tricks nicht nur in M2-Amiga

Prozessieren und Exekutieren (letzter Teil)

Adressen sind für manche Branchen von hoher Bedeutung. Wir Programmierer leben davon. Heute interessieren uns besonders die Adressen von Prozessen.

Um sie anzusprechen beziehungsweise ihnen Nachrichten zuzusenden, erschließen wir uns unter anderem folgende Möglichkeiten:

- Programme abbrechen,
- den Ablauf von Programmen überwachen bzw. fernsteuern,
- Eingaben simulieren,
- Ausgaben umlenken,
- Eigenschaften von Tasks verändern,
- Lage und Größe von Fenstern verändern sowie
- Requesterausgaben beeinflussen.

Eigene Prozeß-adresse entlockt

Am einfachsten fällt es einem Programm, dem Prozeß die Adresse zu entlocken, den es selbst nutzt. Eine einzige Programmzeile genügt:

```
ourProc := ADDRESS(Find
Task(NIL));
```

Die Funktionsprozedur "Exec. FindTask" liefert die Adresse der Task, unter der das aufrufende Programm läuft, wenn als Parameter "NIL" übergeben wird. Die Adresse der Task ist zugleich die Adresse des Prozesses, da Tasks an erster Stelle in den Verbund von Prozeßstrukturen eingefügt sind (vergleiche [1], S. 13-14 oder [2], S. 380 des Literaturverzeichnisses). Die erhaltene Adresse ist vom Typ "TaskPtr". Sie kann nicht ohne weiteres der Variablen "ourProc" (Typ Dos.ProcessPtr) zugewiesen werden. Die beiden Typen sind nicht zuweisungskompatibel. Ein Typentransfer mit "SYSTEM.ADDRESS" beseitigt das Hindernis. Der Typ "ADDRESS" ist zu allen Zeigertypen kompatibel. Korrekter wäre ein Transfer zu dem tatsächlich benötigten Typ. Leider unterstützt der Compiler des M2Amiga den Transfer von Zeigertypen nicht. An dieser Stelle kann jedoch "SYSTEM.CAST" einspringen, das die Typen einfach umschaltet:

```
ourProc := CAST(Pro-
cessPtr,FindTask(NIL));
```

Unabhängig von Ihrer Entscheidung, Typen zu transferieren oder umzuschalten, wird der Code gemäß Listing 1 erzeugt. Damit ist zumindest für das Beispiel erwiesen, daß die strenge Typenbindung von Modula2 nur die Länge des Quelltextes beeinflußt, sich jedoch nicht zur Laufzeit negativ auswirken muß. Mögliche Typenkonflikte deckt bereits der Compiler auf.

Adressen von Workbench-Prozessen

Nun wagen wir uns an fremde Prozesse heran. In der letzten Ausgabe haben wir im Modul "WbCall" die Prozedur "ProzessAnlegen" eingesetzt. Sie ist als Listing 2 abgedruckt. Ausgehend von der "ProcessId", die "CreateProc" liefert, läßt sich die Adresse des neuen Prozesses errechnen. Werfen Sie nochmals einen Blick auf die Prozeßstruktur. Die "ProcessId" ist ein redefinierter "MsgPortPtr", der auf den Message-Port weist, der nach der Task in die Prozeßstruktur eingelagert ist. Somit ist die Adresse des Message-Port lediglich um die Länge der Task-Struktur zu verringern, um die Adresse des Prozesses zu erhalten. Genau das erfolgt in Listing 2.

In den meisten Fällen geht alles glatt. Dennoch sollte man sich davon überzeugen, daß "CreateProc" auch wirklich erfolgreich ablief (zum Beispiel mit Arts.Assert), bevor man mit der Adreßkalkulation beginnt. Dennoch kommt es bei schnelleren Systemen (68020/30) vor, daß die Adreßberechnung bereits abgeschlossen ist, bevor der Prozeß voll eingerichtet wurde. Dieses Problem ist leicht mit "FindTask" zu lösen. Listing 2a zeigt das Vorgehen. In dem Beispiel ist "etwas Sicherheit" eingebaut. Der Aufruf von "FindTask" wurde in eine Schleife gepackt, um sicherzustellen, daß der Prozeß auch wirklich vom Betriebssystem verwaltet wird. In den Looping tritt das Programm nur ein, wenn "CreateProc" erfolgreich war. Folglich gibt es den Prozeß. Wir lassen ihn solange suchen, bis "Exec" sich den Namen entreißen läßt, und verlassen die Schleife.

Und wie verfährt man bei Prozessen, die die Workbench anlegt, wenn ein Programm durch Anklicken von Icons zum Leben erweckt wurde? Der neue Prozeß trägt den Namen des gestarteten Programms. Überzeugen Sie sich mit "XOPER" [3] oder mit "ARTM" [4] davon. Somit kann "FindTask" auch hier "zuschlagen".

Adresse eines "Background CLI" erschleichen

Bleibt uns noch der Fall "Meier", der Prozeß mit dem Allerweltsnamen "Background CLI". Erinnern Sie sich noch an die Ausführungen in der Ausgabe 5/90? Wir stellten damals fest, daß mit "Dos.Execute" angelegte Prozesse stets den Namen "Background CLI" tragen. Wir stoßen nun an die Grenzen der Systemroutine "FindTask", die uns nur die Adresse eines Prozesses (einer Task) liefern kann. Ist es die von uns gesuchte? Selbst ein wiederholter Aufruf von "FindTask" muß nicht zum gewünschten Ergebnis führen, weil die vom Betriebssystem angelegten und verwalteten Listen der Tasks stets von vorne durchsucht werden. Genug gejammert, her mit der Lösung!

Ein einfacher Weg, die "Meiers" zu überlisten, besteht darin, sämtliche bestehenden "Background CLI" vor dem Aufruf von "Dos.Execute" in "Müller" umzutaufen. Damit sind die Voraussetzungen geschaffen, daß "FindTask" auf Anhieb das benötigte Ergebnis liefern kann. Als fairer Nutzer des Betriebssystems sollte man die "Müllers" schnell wieder als "Meiers" umstempeln, wenn "FindTask" seine Aufgabe abgeschlossen hat. Die beschriebenen Arbeiten leistet die Routine "AendereTaskName" des Listing 3. In dem Namen wird das Leerzeichen zwischen "Background" und "CLI" geändert. Mehr ist zur Unterscheidung nicht notwendig. Sie könnten auch anders verfahren, sollten jedoch einen Namen wählen, der die Länge der Zeichenkette "Background CLI" nicht überschreitet.

Die neue Routine könnten Sie zum Beispiel mit dem Modul "BackgroundCLI", Listing 4, erproben und sich mit "ARTM" von der Funktionsfähigkeit überzeugen. Das Modul zeigt auch, wie es sicher möglich ist festzustellen, ob der mit "Dos.Execute" angelegte Prozeß auch wirklich existiert. Auf die Routine "ProzessAbbrechen", die von der Prozeßadresse Gebrauch macht, gehen wir später ein.

Slot-Maschine

Das Betriebssystem (Dos) führt in der Datenstruktur "Dos.TaskArray" (siehe [1], Seite 13-16) über die CLI-Prozesse Buch, das heißt über Prozesse, die mit Mitteln des "Dos" angelegt wurden. Ausgehend von der Arbeitsweise des "Dos", einen neu angelegten CLI-Prozeß in die erste freie Position des "TaskArray" einzutragen, ist es uns nun ein leichtes, an die Adresse des Prozesses heranzukommen. Vor dem Aufruf von "Dos.Execute..." ermitteln wir die erste freie Position, lesen später die "ProcessId" heraus und berechnen die Adresse, wie bereits in Listing 2 praktiziert. In dem Modul "TaskArray" (Listing 5) testen wir unser Vorgehen. Damit ist ein Weg eröffnet, auch ohne Umbenennung von Prozessen auszukommen.

Tauchen wir nun tiefer in Listing 5 ein: Das Programm ist nur in einer CLI-Umgebung lauffähig, da ein Prozeß-Window vorausgesetzt wird (3.

Parameter von Dos.Execute = NIL). Durch Test der Variablen "Arts.wbStarted" ist die Voraussetzung garantiert.

Es gilt nun den ersten freien Slot (Schlitz), das heißt die Adresse des ersten Speicherplatzes innerhalb des "TaskArray" mit dem Eintrag "NIL" festzustellen. Bei Verwendung von "Dos.Execute" ohne "run" im ersten Parameter hätten wir dann die Adresse, an der das Betriebssystem die gesuchte Referenz auf den angelegten Prozeß einträgt. Unser Fall liegt jedoch ein wenig anders, weil wir "run" im ersten Parameter verwenden, um einen asynchronen Prozeß zu kreieren. Dieses Vorgehen hat zur Folge, daß erst der zweite freie Slot die von uns gesuchte "ProcessId" aufnimmt. Der erste freie Slot wird nur temporär genutzt. Nun ist auch die Funktion der Routine "UebernaechsterSlot" verständlich.

"UeberNaechsterSlot" ist aufzurufen, bevor der asynchrone Prozeß angelegt wird. Als Ergebnis wird ein Zeiger auf das Element (Langwort) des "TaskArray" geliefert, in dem das Betriebssystem die "ProcessId" vermerkt. Nach Start des Prozesses mit "Dos.Execute"(ADR"run...") muß etwas Zeit (Dos.Delay(50)) verstreichen, bis die "ProcessId" eingetragen ist. (Alternativ zur gewählten Verzögerung hätte auch mit einem "LOOP" sinngemäß zu Listing 2a gearbeitet werden können.) Die Adresse des Prozesses wird entsprechend Listing 2 errechnet.

Handbremse anziehen

Asynchrone Prozesse sollten sich von selbst terminieren oder von außen abbrechbar sein. Die Routine "ProzessAbbrechen" (Listing 4) tut genau das, was ihr Name verheißt. Aus der Prozeßstruktur wird die Nummer des CLI-Prozesses gewonnen, in ein ASCII-Zeichen umgewandelt und in die Zeichenkette "abbruch" eingebracht, dem ersten Parameter bei dem Aufruf von "Dos.Execute". Wir lassen den CLI-Befehl "C:break" für uns erfolgreich arbeiten. Wenn es funktioniert, ist auch der Beweis erbracht, daß wir die richtige Prozeßstruktur zu fassen bekommen haben. Haben Sie schon einmal ein Programm mit "CTRL-C" abgebrochen? Der manuelle Vorgang wird in Listing 6 durch Übermittlung des Signals mit der Nummer 12 ersetzt, das als "Exec.ctrlC" deklariert ist.

Notbremse

Warum fassen wir eigentlich Prozesse mit Samthandschuhen an und praktizieren den "Rausschmiß" nicht mit "Exec.RemTask"? Auch dieser Weg wäre gangbar, quasi als Notbremse. Dem Laufzeitsystem wäre dann die Chance genommen, Ressourcen ordnungsgemäß an das Betriebssystem zurückzumelden. Wir hätten nachzuarbeiten, zum Beispiel Fenster und Dateien zu schließen.

Um die Dinge praktisch zu erproben, starten Sie bitte den Compiler "m2c" von der Workbench und anschließend das Miniprogramm "KillM2c" (Listing 7). Das Fenster mit dem Titel "m2c" bleibt geöffnet. Mit dem "ARTM" könnten Sie feststellen, daß auch der Message-Port "m2c" weiterhin besteht. Auch der Arbeitsspeicher bleibt mit dem Code des Moduls "m2c" belegt. Weiterhin könnten Sie alternativ zu "Exec.RemTask" Listing 6 in "KillM2c" einbauen, um den Unterschied selbst zu erfahren.

Es bedarf nicht immer eines Tochterprozesses, um aus einem Programm heraus ein anderes zu starten. Insbesondere, wenn auf Ergebnisse gewartet werden muß, also bei synchronen Abläufen, bietet es sich an, "konventionelle Mittel" einzusetzen und auf ohnehin nicht genutzte Möglichkeiten des Multitasking zu verzichten.

Recht auf CPU ohne Process & Execute

In Listing 8 ruft das Modul "CliStart" den Compiler auf, um sich selbst zu übersetzen. Das Modul ist lauffähig. Es setzt voraus, daß das aktuelle Verzeichnis auf das Projekt gerichtet ist, welches "CliStart" enthält.

Die eigentliche Arbeit leistet die Prozedur "StartModule", die als Parameter zwei Zeiger auf Zeichenketten erwartet, auf den Modulnamen (nebst Pfad) und auf etwaige Argumente. Die Parameter übergeben wir in Registern (zur Technik siehe Amiga Dos 1/90, Seite 68). Die Wahl fiel nicht zufällig auf d1 und a0. Beim Laden des Moduls (Dos.LoadSeg) wird der Zeiger auf dem Modulnamen in d1 erwartet. Beim Start eines Programms werden die Parameter gemäß Amiga-Programmierkonvention in a0 und d0 vorausgesetzt (siehe dazu ausführlich [2], Seite 389 oder [5], Seite 594).

Nichts geht ohne Liste

Das Betriebssystem verwaltet die einzelnen Codeblocks eines Programms in einer Segmentliste ([2], Seite 356), deren Definition nicht Bestandteil der Schnittstellenmodule ist. Folglich mußten wir den Datentyp selbst definieren. Die Elemente der Liste sind über "next" verknüpft. Uns interessiert das Element "code". Hier beginnt jeweils das erste Byte eines Codeblocks. Die Länge wurde willkürlich angegeben. Eine Variable dieses Typs wird nicht angelegt, somit auch kein Arbeitsspeicher dafür beansprucht. Wir benötigen lediglich den Zeiger auf das erste Listenelement, genauer auf den Anfang des Codes. Den Zugriff auf das erste Segment erhalten wir mit:

```
segment := LoadSeg(module);
```

Mit "ADR(segment↑.code)" haben wir den Anfang des Programms ermittelt.

Prozedurvariable, ein Vielkönner

Und wie soll das geladene Modul gestartet werden ? Nur in Assembler scheint das mit einem "jsr (address)" ganz einfach zu sein. Modula ermöglicht den Sprung auf gleiche Weise mit Hilfe von Prozedurvariablen, in unserem Fall mit "callModule" vom Typ "ProgAufruf". Die Variable arbeitet mit einem Parameter des Typs LONGINT. Eine Prozedurvariable ist wie ein Prozeduraufruf zu verstehen. Parameter werden normalerweise auf dem Stack übergeben. Und genau da erwartet das Betriebssystem die Länge des verfügbaren Stack bei Aufruf eines Programms ([2], Seite 388). Wir können somit mühelos den verfügbaren Stack als Parameter bei Aufruf der Prozedurvariablen übergeben.

Um Programmieraufwand zu sparen, haben wir vereinfachend angenommen, daß 200 Byte des Stacks (des Prozesses) bereits belegt sind. Das Laufzeitsystem "Arts" hält für uns die Stacklänge bereit. Wir hätten sie auch direkt der Prozeßstruktur entnehmen können.

Den Modulstart bereiten wir durch

```
CallModule :=
ADR(segment↑.code);
```

vor und mit

```
returnCode := CallModu-
le(stackSize - 200);
```

geht es dann los. Es erfolgt ein "jsr (a3)" zum zu startenden Modul. Wir haben somit Register wie folgt eingesetzt:

a0 :	Zeiger auf Zeichenkette mit Argumenten
d0 :	Länge der Zeichenkette
a3 :	Adresse des geladenen Codes
sp :	Stacklänge

Elegant, oder? Schauen Sie sich doch einmal die Codegenerierung mit "DisObject" [6] an.

Den "returnCode" wollen wir auch noch kurz besprechen. Alle unseren bisherigen Bemühungen, die echten Return-Codes von Programmen zu erhalten, scheiterten z.B. an der Arbeitsweise von "Dos.Execute". Bei direktem Start von Programmen, wie in Listing 8, erhalten wir entsprechend der Amiga-Programmierkonventionen Rückgabewerte in d0, somit in der Variablen "returnCode".

Workbench zieht gleich

Das Programm gemäß Listing 8 arbeitet nur bei Start vom CLI. Mit dem Wissen aus AMIGA DOS 6/90 können wir schnell eine Lösung für die Workbench schaffen. Listing 9 zeigt das Vorgehen. Im wesentlichen ist die Prozeßstruktur um eine teilweise initialisierte CLI-Struktur zu ergänzen. Weiterhin ist ein Ausgabefenster bereitzustellen. Versäumen Sie es nicht, den Stack im Icon für das Modul "WorkBenchCliStart" auf etwa 15000 zu setzen, um dem Compiler die notwendigen Arbeits-

bedingungen zu schaffen, oder unterlassen Sie es, um zu testen, ob der Return-Code wirklich bereitgestellt wird.

Anregung

Die in der heutigen Folge dargestellten Wege, Programme zu starten, sind noch lange nicht erschöpft. Wir haben Ihnen Möglichkeiten aufgezeigt, auf asynchrone Prozesse zuzugreifen und synchrone durch eine "Unterprogrammtechnik" zu ersetzen. Was aber bleibt zu tun, wenn die Länge des Stacks eines laufenden Prozesses variabel gehalten werden soll? Es bietet sich an, das Coroutinen-Konzept einzusetzen. Die Nebenläufigkeit interessiert uns in diesem Zusammenhang weniger, wohl aber die Möglichkeit, der Taskstruktur je nach Bedarf einen beliebigen Stack zu verpassen. Schicken Sie uns Ihre Lösung, insbesondere wenn Sie mit mehreren Programmiersprachen realisiert werden kann. Arbeiten Sie doch einmal mit dem Modul "Coroutines". In [7] werden interessante Anwendungen aufgezeigt.

(mm)

Literatur:

[1] M2Amiga Handbuch, 1989

[2] Das Amiga Guru-Buch, 1989

[3] Fred Fish Disk Nr. 274

[4] Fred Fish Disk Nr. 277

[5] AMIGA ROM Kernel, Libraries & Devices, 1989

[6] PD-Disk Amok Nr. 12

[7] Programmierung in modula2, 1989

Listings






```
VAR ourProc : ProcessPtr;
BEGIN
...
(* Typenkonvertierung *)
(* ourProc := ADDRESS(FindTask(NIL)); *)
(Typenumschaltung *)
ourProc := CAST(ProcessPtr,(FindTask(NIL)));
MOVE.W  #0,A1       ; Setze Parameter NIL
MOVE.L  -56(PC),A6  ; hole Adr Exec-Lib
JSR     -294(A6)    ; FindTask
MOVE.L  D0, -8(A4)  ; Ergebnis nach ourProc
```

```
VAR ourProc : ProcessPtr;
PROCEDURE ProzessAnlegen (VAR stMsg : WBStartup);
BEGIN
  WITH stMsg DO
    process := CreateProc(ADR(ProcName),2,
                          segment,30000);
    (* Prozessadresse errechnen *)
    ourProc := ADDRESS(process);
    DEC(ourProc,TSIZE(Task));
  END;
END ProzessAnlegen;
```

```
VAR ourProc : ProcessPtr;
PROCEDURE ProzessAnlegen (VAR stMsg : WBStartup);
BEGIN
  WITH stMsg DO
    process := CreateProc(ADR(ProcName),2,
                          segment,30000);
    Assert(process # NIL, ADR("kein Prozess"));
  END;
  LOOP
    ourProc := ADDRESS(FindTask(ADR(ProcName)));
    IF ourProc # NIL THEN
      EXIT
    END;
  END;
END ProzessAnlegen;
```

```
CONST BackgroundName = "Background CLI";
PROCEDURE AendereTaskNamen(VAR name:ARRAY OF CHAR;
                               zeichen : CHAR);
VAR ourTask : TaskPtr;
    namePtr : POINTER TO ARRAY[0..15] OF CHAR;
    gesucht : ARRAY[0..15] OF CHAR;
BEGIN
  Str.Copy(gesucht, name); (* Ziel, Quelle *)
  LOOP
    ourTask := FindTask(ADR(gesucht));
    IF ourTask # NIL THEN
      namePtr := ourTask^.node.name;
      (* Background CLI *)
      (*           ^    *)
      (*       zeichen *)
      namePtr^[10] := zeichen;
      Str.Copy(name, namePtr^);
    ELSE
      EXIT
    END;
  END;
END AendereTaskNamen;
```

Listing: Modula-Tricks

```
MODULE BackgroundCLI;
FROM Arts IMPORT
        BreakPoint, wbStarted;
FROM Dos IMPORT
        Delay, Execute, ProcessPtr;
FROM Exec IMPORT
        FindTask, TaskPtr;
FROM Str IMPORT
        Copy, Length;
FROM SYSTEM IMPORT
        ADDRESS, ADR;
FROM Terminal IMPORT
        Write, WriteLn, WriteString;
CONST BackgroundName = "Background CLI";
VAR taskName : ARRAY[0..15] OF CHAR;
    ourProc  : ProcessPtr;
    done     : LONGINT;
PROCEDURE AendereTaskNamen(VAR name:ARRAY OF CHAR;
                               zeichen : CHAR);
(* bitte aus Listing 3 kopieren *)
END AendereTaskNamen;
PROCEDURE CommandNameAusgeben(proc:ProcessPtr);
VAR
   namePtr : POINTER TO ARRAY[0..15] OF CHAR;
BEGIN
  WriteString("Prozessname   : ");
  WITH proc^ DO
    namePtr := task.node.name;
    WriteString(namePtr^); WriteLn;
    namePtr := ADDRESS(cli^.commandName);
    INC(namePtr);
    WriteString("Programmname : ");
    WriteString(namePtr^); WriteLn;
  END;
END CommandNameAusgeben;
PROCEDURE ProzessAbbrechen(proc : ProcessPtr);(*PA*)
VAR procNummer : CHAR;
    abbruch    : ARRAY[0..21] OF CHAR;
    laenge     : LONGCARD;
BEGIN
  Delay (150);
  procNummer := CHAR(proc^.taskNum + 30H);
  abbruch := "run c:break ";
  laenge := Length(abbruch);
  abbruch[laenge] := procNummer;
  abbruch[laenge + 1] := 0C;
  done := Execute(ADR(abbruch), NIL, NIL);
END ProzessAbbrechen;
BEGIN
  IF ~ wbStarted THEN
    taskName := BackgroundName;
    AendereTaskNamen(taskName,"*");
    done := Execute(ADR("run m2:m2c"),NIL,NIL);
    (* Es werden zwei Prozesse angelegt und der
       erste wieder aufgehoben. Das kostet Zeit. *)
    Delay
    ourProc := ADDRESS(FindTask(ADR(BackgroundName)));
    CommandNameAusgeben(ourProc);
    AendereTaskNamen(taskName," ");
    ProzessAbbrechen(ourProc);
  END;
END BackgroundCLI.
```

```
MODULE TaskArray;
FROM Arts IMPORT
        BreakPoint, wbStarted;
IMPORT Dos;
FROM Dos IMPORT
        Delay, DosLibraryPtr, Execute, ProcessPtr, Tas
kArrayPtr;
FROM Exec IMPORT
        Task;
FROM Str IMPORT
        Copy, Length;
FROM SYSTEM IMPORT
```

Listing: Modula-Tricks

```
        ADDRESS, ADR, SETREG, TSIZE;

VAR
    ourProc  : ProcessPtr;

PROCEDURE ProzessAbbrechen(proc : ProcessPtr);

(* Bitte aus Listing 4 kopieren *)

END ProzessAbbrechen;


TYPE TaskSlotPtr = POINTER TO LONGINT;

PROCEDURE UebernaechsterSlot() : TaskSlotPtr;
VAR dosLib  : DosLibraryPtr;
    slotPtr : TaskSlotPtr;

PROCEDURE NaechsterSlot(VAR sltPtr : TaskSlotPtr);
BEGIN
  WHILE sltPtr^ # NIL DO
    INC(sltPtr, 4);
  END;
END NaechsterSlot;

BEGIN
  dosLib := ADR(Dos);
  slotPtr := ADDRESS(dosLib^.root^.taskArray);
  NaechsterSlot(slotPtr);
  INC(slotPtr, 4);
  NaechsterSlot(slotPtr);
  RETURN slotPtr;
END UebernaechsterSlot;

VAR expectedSlot : TaskSlotPtr;

BEGIN
  IF ~ wbStarted THEN
    expectedSlot := UebernaechsterSlot();
    SETREG(0, Execute(ADR("run m2:m2c"),NIL,NIL));
    (* Es werden 2 Prozesse angelegt und der erste
       wieder entfernt. Das kostet Zeit.*)
    Delay(50);
    ourProc := ADDRESS(expectedSlot^);
    DEC(ourProc, TSIZE(Task));
    ProzessAbbrechen(ourProc);
  END;
END TaskArray.
```

```
PROCEDURE ProzessAbbrechen(ourTask : ADDRESS);

BEGIN
  Delay (50); (* zur Sicherheit *)
  Signal(ourTask, LONGSET{ctrlC});
END ProzessAbbrechen;
```

```
MODULE KillM2c;

FROM Arts IMPORT Assert;
FROM Exec IMPORT FindTask, RemTask, TaskPtr;
FROM SYSTEM IMPORT ADR;


VAR killTask : TaskPtr;

BEGIN
  killTask := FindTask(ADR("m2c"));
  Assert(killTask # NIL, ADR("m2c nicht da"));
  RemTask (killTask);
END KillM2c.
```




```
MODULE CliStart;

(* $F- $R- $S- $V- $Z- *)

FROM Arts    IMPORT stackSize;
FROM ASCII   IMPORT lf;
FROM Dos     IMPORT LoadSeg;
FROM InOut   IMPORT
             WriteInt, WriteLn, WriteString;
FROM Str     IMPORT  Length;
FROM SYSTEM  IMPORT
             ADDRESS, ADR, BYTE, SETREG;

TYPE STRING = ARRAY[0..49] OF CHAR;
     STRPTR = POINTER TO STRING;

CONST a0 = 10B; d0 = 0B; d1 = 1B;

VAR aufzustartendesModul,
    modulParameter         : STRING;
    modulErgebnis          : LONGINT;

PROCEDURE StartModule(module {d1},
                      arguments {a0} : STRPTR)
                      : LONGINT;

TYPE ProgAufruf = PROCEDURE(LONGINT) : LONGINT;
     Segment    = RECORD
       next : BPOINTER TO Segment;
       code : ARRAY[0..99] OF BYTE; (* dummy *)
     END;

VAR callModule : ProgAufruf;
    segment    : BPOINTER TO Segment;
    laenge     : LONGINT;
    returnCode : LONGINT;

BEGIN
  segment := LoadSeg (module);
  IF segment # NIL THEN
    callModule := ADR(segment^.code);
    laenge := Length(arguments^);
    arguments^[laenge] := lf;
    SETREG(d0, laenge);
    returnCode := callModule(stackSize - 200);
  END;
  RETURN returnCode;
END StartModule;

BEGIN
  aufzustartendesModul := "m2:m2c";
  modulParameter := "CliStart";
  modulErgebnis := StartModule
                   (ADR(aufzustartendesModul),
                    ADR(modulParameter));
  WriteString("Compiler Ergebnis-Code : ");
  WriteInt(modulErgebnis, 3); WriteLn;
END CliStart.
```

Listing: Modula-Tricks

```
MODULE WorkBenchCliStart;

(* $F- $R- $S- $V- $Z- *)

FROM Arts    IMPORT stackSize, wbStarted;
FROM ASCII   IMPORT lf;
FROM Dos     IMPORT
             Close, CommandLineInterface, LoadSeg,
             newFile, Open, ProcessPtr, UnLoadSeg;
FROM Exec    IMPORT FindTask;
FROM InOut   IMPORT
             WriteInt, WriteLn, WriteString;
FROM Str     IMPORT Length;
FROM SYSTEM  IMPORT
             ADDRESS, ADR, BPTR, BYTE, SETREG;

TYPE STRING = ARRAY[0..49] OF CHAR;
     STRPTR = POINTER TO STRING;

CONST a0 = 10B; d0 = 0B; d1 = 1B;


VAR aufzustartendesModul,
    modulParameter         : STRING;
    modulErgebnis          : LONGINT;
    ourProcess             : ProcessPtr;
    cliStruktur            : CommandLineInterface;


PROCEDURE StartModule(module {d1},
                      arguments {a0} : STRPTR)
                      : LONGINT;

(* bitte aus Listing 8 uebernehmen *)

END StartModule;

PROCEDURE ProcessPatchen (proc : ProcessPtr);
BEGIN
  WITH proc^ DO
    cis := Open(ADR("CON:0/0/400/200/ WBStart "), newF
ile);
    cos := cis;
    cli := BPTR(ADR(cliStruktur));
    WITH cli^ DO
      standardInput  := cis;
      currentInput   := cis;
      standardOutput := cos;
      standardOutput := cos;
      defaultStack := (stackSize - 200) DIV 4;
    END;
  END;
END ProcessPatchen;


BEGIN
  IF wbStarted THEN
    ourProcess := ADDRESS(FindTask(NIL));
    ProcessPatchen(ourProcess);
  END;

(* bitte hier Modulrumpf aus Listing 8 einfuegen *)

  IF wbStarted THEN
    Close(ourProcess^.cis);
  END;

WorkBenchCliStart.
```

Listing: Modula-Tricks

Gewußt wie!

Die Seiten für Einsteiger und Fortgeschrittene

Hier sind sie wieder, unsere Kurztips für das schnelle Nachschauen. Ob CLI oder BASIC, Assembler oder Workbench – hier finden Ratsuchende Hilfe. Doch nun haben unsere Leser das Wort.

Rund um den Amiga

Hier ein Tip für alle, die mit ihrem Amiga längere Berechnungen durchführen (zum Beispiel bei Animationsprogrammen, Graphiken, Ray-Tracing, Crunchern und so weiter):
Wenn man über eine längere Zeit seinen Amiga etwas berechnen läßt, spart man Zeit, indem man einfach den Monitor abschaltet. Dies wird vom Amiga erkannt, und er hat mehr RAM für das Berechnen!
Hier ein kleines Programm in Amiga-BASIC, mit dem Sie die Koordinaten der Maus abfragen können:

```
wind&=WINDOW(7)
WHILE 1
   LOCATE 1,1 'Hierher schreibt das Programm die Koordinaten!
   PRINT PEEKW(wind&+14),PEEKW(wind&+12)
WEND
```

Eine Zentrierung für die Textausgabe in Amiga-BASIC läßt sich so realisieren:

```
SUB ZENTR (a$) STATIC
  POKEW WINDOW(8)+36,WINDOW(2)/2-LEN(a$)*4
  PRINT a$
END SUB
```

Der Aufruf lautet demnach:

```
ZENTR "Dieser Text steht ganz genau"
ZENTR ">>>>>> IN DER MITTE ! <<<<<<"
```

(Kai Bühler/jb)

Backup-Test (GFA-BASIC)

Wenn man im GFA-BASIC-Editor Programme abspeichert, prüft der Interpreter, ob das zu speichernde File schon existiert und legt in diesem Fall ein Backup-File an, das dem letzten Stand des gespeicherten Files entspricht. Dies ist sehr nützlich, wenn man einen Fehler gemacht hat und noch einmal den Stand vor der letzten Änderung abrufen möchte.
In eigenen Programmen wird nun diese Backup-Abfrage vom Interpreter nicht durchgeführt, wenn man Files mit »OPEN ...« öffnet und Daten speichert.
Es wäre aber durchaus nützlich, wenn man zum Beispiel in einer Dateiverwaltung immer noch den letzten Stand abrufen könnte. Dazu bindet man einfach die Funktion »BACKUP« in seine Programme ein:
Bevor ein File mit »OPEN« geöffnet wird, wird die Funktion aufgerufen, und man erhält einen Wert, der anzeigt, ob ein Backup-File angelegt wurde oder nicht; der Aufruf:

```
antwort! = @backup(file$)
```

oder

```
antwort! = FN backup(file$)
```

Man erhält entweder »TRUE«, – das heißt, »file$« existiert schon und Backup wurde angelegt, oder »FALSE«, – »file$« existiert nocht nicht, kein Backup angelegt. Wenn man an der Anwort der Funktion nicht interessiert ist, kann man auch einfach den VOID-Befehl des GFA-Interpreters benutzen.

```
VOID @backup(file$)
```

führt die Funktion aus, liefert aber keine Antwort.

Funktionsweise: Die Funktion »BACKUP(FILE$)« untersucht, ob das zu speichernde File schon existiert. Der Name des Files ist in »file$« abgelegt und kann auch einen vollen Pfad enthalten. Existiert das File schon, wird das eventuell auch schon vorhandene Backup-File gelöscht und das gespeicherte File in »FILE$.bak« umbenannt. Danach kehrt die Funktion zum Aufruf zurück, und das File kann geöffnet werden, um die Daten zu speichern.
Das eventuell schon vorhandene Backup-File muß gelöscht werden, da sonst GFA-BASIC beim Versuch, die alte Datei umzubenennen, die Fehlermeldung: »Objekt existiert schon« ausgibt.
Hier das Programm:

```
' % BACKUP-TEST %
' % kommentiert %
' % B.Funke %
'
FUNCTION backup(file$)
IF EXIST(file$)=TRUE ! falls zu speicherndes File schon exi-
  stiert
IF EXIST(file$+".bak")=TRUE ! falls Backup-File schon
  existiert:
KILL file$+".bak" ! Backup-File löschen (sonst Fehler-
ENDIF ! meldung: Objekt existiert schon)
NAME file$ AS file$+".bak" ! zu speicherndes File in Backup-
  File
RETURN TRUE ! umbenennen und TRUE zurückgeben
ENDIF
RETURN FALSE ! sonst FALSE zurückgeben
ENDFUNC
```

(Boris Funke/jb)

Gerätenamen in Lade-Requestern

In einigen Programmen finden sich in den Lade-Requestern Gadgets wie zum Beispiel » DH0:«, »DF2:«, »DF3:«. Diese mögen ja für die, die diese Geräte auch besitzen, ganz nützlich sein. Für alle anderen ist folgender Tip ganz hilfreich. Bei Deluxe-Print II zum Beispiel findet sich im Lade-Requester das Gadget »DH0:«. Diejenigen Amiga-User, die keine Festplatte besitzen, können dieses Gadget normalerweise vergessen. Aber man kann das Gadget auch für ein anderes Gerät verwenden, indem man auf der Deluxe-Print-Diskette im Verzeichnis »Language« das File »Deutsch« sucht und mit einem Editor lädt. In diesem File befindet sich der deutsche Text für das Programm. Dort sucht man nach dem Eintrag «DH0:«. Diesen kann man dann durch einen anderen ersetzen (zum Beispiel »DF2: RAM: VD0: CARD:«), und schon greift das Programm beim Anklicken des Gadgets auf das gewünschte neue Gerät zu. Das funktioniert ähnlich bei Deluxe-Paint II, wobei man das Programm mit einem File-Monitor (zum Beispiel »NewZap« aus dem Public-Domain-Pool oder »FileMon« aus Heft 5/90) lädt und nach dem zu ersetzenden String (zum Beispiel »DF0:«,»DF1:« und so weiter) sucht. Dieser wird dann durch den gewünschten Gerätenamen ersetzt. Bei Änderungen des Programmcodes mit »NewZap« sollte man dies unbedingt mit einer Kopie des Programms machen und nicht mit dem Original! So ist das Programm nicht verloren, wenn mal etwas

schief geht. Außerdem empfiehlt es sich, das zu ändernde Programm in eine RAM-Disk zu laden, da dort das Laden und das Durchsuchen des Programms nach dem gewünschten String wesentlich beschleunigt wird, was sich besonders bei langen Programmen wie Deluxe-Paint bemerkbar macht. Bei welchen Programmen diese Methode zum Erfolg führt, muß man im einzelnen ausprobieren.

(Norbert Dimpfl/jb)

Sprechen mit dem Speaker!

In der Ausgabe AMIGA DOS 2'90 wird beschrieben, wie man die Sprachausgabe des Amiga vom CLI aus nutzen kann.

Mit der Workbench 1.3 läßt sich dieses noch vereinfachen. In der Mountlist der Workbench 1.3 steht der Eintrag des Gerätes »SPEAK:«. Dieses müssen wir nur mit »MOUNT SPEAK:« anmelden und sind dann in der Lage, die Sprachausgabe zu nutzen. Wir können uns damit zum Beispiel die Startup-Sequence vorlesen lassen, indem wir einfach »COPY s:startup-sequence to speak:« eingeben. Oder wir lassen uns vom Amiga begrüßen. Dazu erstellen wir ein File mit dem entsprechenden Text und speichern dieses auf der Workbench. Die Startup-Sequence erweitern wir folgendermaßen:

```
...
Mount Speak:
copy sys:<text.file> to ram:
run copy ram:<text.file> to speak:
...
```

Durch das Kopieren des Textfiles ins RAM verhindern wir ein Kollidieren der Sprachausgabe mit der Abarbeitung der Startup-Sequence, da beide auf die Bootdiskette zugreifen würden, was das berühmte "Sägen" zur Folge hätte. Außerdem hat das Kopieren den Vorteil, daß wir mit dem Befehl »RUN COPY...« die Sprachausgabe als Hintergrund-Task starten können. So wird ermöglicht, daß trotz der Sprachausgabe die Startup-Sequence weiter abgearbeitet werden kann und der Bootvorgang nicht wesentlich länger dauert. Auch das Umleiten einer Ausgabe ist möglich. So können wir uns zum Beispiel ein Verzeichnis einer Diskette vorlesen lassen:

```
dir >Speak: df1:[Verzeichnis/Verzeichnis/...]
```

Der Vorteil des Gerätes »SPEAK:« ist der, daß man sich um die Übersetzung des Textes in Phoneme nicht zu kümmern braucht, da das Gerät dies über die »TRANSLATOR.LIBRARY« selbst erledigt. Ein Nachteil soll aber nicht verschwiegen werden. Durch die Übersetzung über die Library ist es schwierig, den »SPEAK:« verständliche Sätze in Deutsch sprechen zu lassen.

Auf der Workbench müssen folgende Files verfügbar sein: »C:MOUNT«, »DEVS:MOUNTLIST« (mit dem Eintrag für das Gerät SPEAKER:), »DEVS:NARRATOR.DEVICE«, »LIBS:TRANSLATOR.LIBRARY«, »L:SPEAK.HANDLER«.

Diese Files gehören aber normalerweise zur "Standardausrüstung" der Workbench 1.3.

(Norbert Dimpfl/jb)

Tip zum Nullmodem

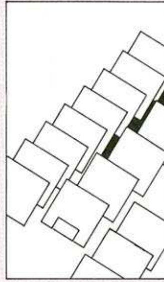
Amiga-500-Besitzer haben es vielleicht schon bemerkt, einige Steckverbindungen sind nicht ganz normgerecht belegt. Dies gilt für Centronics-(Parallel-)Port, aber auch die Verwendung von "male"-Steckern rechnerseitig ist ungewöhnlich. Gravierend wird das zum Beispiel, wenn man versucht, ein RS-232-Kabel zu löten, und nicht sehr gut aufpaßt. Die Belegung des anzulötenden 25poligen "female"-Steckers ist dabei wie folgt (mit Sicht auf Rechner beziehungsweise Lötpins des Steckers):

```
01...........13
 14..........25
```

Die Belegung der Kontakte erfolgt dann entsprechend der Numerierung, wie sie in der AMIGA DOS 5/90 gezeigt wurde. Die andere Seite des Kabels kann beim PC oder vielen anderen Rechnern wie angegeben erfolgen, man sollte sich aber auch hier im jeweiligen Rechner-Manual vergewissern, ob und wie die Pins belegt sind. Beim Amiga 500 hätte eine andere Belegung gravierende Folgen, da die Pins 9 und 10 Spannung führen, die bei einer Belegung, wie sie im oben genannten Artikel gezeigt wurde, auf die Leitungen RTS und CTS des anderen Rechners gelangen würde.

Nun noch ein Tip zu Übertragungsproblemen: Es ist möglich, nur den Taktgenerator eines Rechners zu benutzen. Dies eröffnet auch die Möglichkeit, Meßgeräte oder ähnliches auszulesen. Hierzu wird in jedem Stecker Pin 4 mit 5 verbunden, wobei die Verbindung durch das Kabel unterbrochen beziehungsweise gar nicht erst hergestellt wird. Alle anderen Leitungen werden wie üblich verbunden, dann der Pin 24 des Rechners (Taktgenerator) mit den Pins 15 und 17 des anderen Gerätes.

(Dirk Bublitz/jb)



Tausche Tip gegen DATABOX

Haben Ihnen die Kurztips auf unseren Seiten gefallen? Vielleicht haben Sie selbst einen Tip parat, der andere Amiga-Besitzer interessieren könnte? Dann machen wir Ihnen wieder unser Angebot: Sie schicken einen oder mehrere kurze Tips an uns, und wenn uns diese gefallen, bekommen Sie von uns im Austausch die DATABOX des Heftes zugeschickt, in dem Ihr Tip erscheint.

Dazu gibt es jetzt noch etwas mehr:

Die drei besten Tips werden mit je 100, – DM belohnt.

Die Redaktion der AMIGA DOS sucht ab sofort unter allen Einsendern die drei **Supertips des Monats** aus, die außer der DATABOX noch je einhundert deutsche Märker (100, – DM) bekommen.

Dabei sollten Sie noch folgendes beachten:

– *Schreiben Sie Ihren Tip als ASCII-Datei auf Ihre Diskette, möglichst ohne eigensinnige Verschnörkelungen, da alle Texte zur weiteren Verarbeitung auf Personalcomputer transferiert werden müssen. Sonder- und Steuerzeichen haben dabei meist seltsame Auswirkungen.*

– *Schreiben Sie Ihre lückenlose Anschrift sowie einen kleinen Hinweis, was auf der Diskette zu finden ist, auf den Brief und auf einen kleinen Aufkleber auf der Diskette.*

Dann können Sie sicher sein, daß die DATABOX Sie erreicht. Schicken Sie das Ganze an den

DMV-Verlag
Redaktion AMIGA DOS
Kennwort "Tausche Tip gegen DATABOX"
Postfach 250
D-3440 Eschwege

Die drei besten Tips des Monats

Hier sind sie nun, die ersten "drei Tips des Monats". Machen Sie mit. Jeder Tip des Monats wird mit 100, – DM belohnt. Also ran an den Amiga und probiert, wie man Einsteigern das Computerleben erleichtert.

CLI IN BASIC!

Ist während der Arbeit mit BASIC ein CLI-Fenster geöffnet, hat man die Möglichkeit, CLI-Befehle in BASIC auszuführen:

```
DECLARE FUNCTION Execute& LIBRARY
LIBRARY "dos.library"
Befehl$="echo AMIGA_DOS_IST_SUUUPER_!!!"
Befehl$=Befehl$+CHR$(0) :REM Beispielbefehl
Status%=Execute& (SADD(Befehl$),0,0)
```

Sofern ein CLI-Fenster empfangsbereit ist, wird in diesem der Text "AMIGA_DOS_IST_SUUUPER_!!!" ausgegeben. Auch ein multitaskingfähiges Programm kann gestartet werden.

```
Befehl$="SYS:Preferences"+CHR$(0)
```

Nach Beenden von Preferences fährt Amiga-BASIC fort, das Programm zu bearbeiten.

(Hans Holger Rutz/jb)

Der Ersatz für den PAINT-Befehl

Vor einiger Zeit war ich damit beschäftigt, ein Zeichenprogramm in BASIC zu entwickeln. Bis auf die Füllfunktion ging auch alles glatt: Der BASIC-Befehl »PAINT« allerdings arbeitet nur im Outline-Modus. Dabei passiert folgendes: Zeichnet man ein gefülltes schwarzes Dreieck, drumherum einen weißen Rand und füllt das Dreieck weiß, so wird nicht nur das Dreieck gefüllt, sondern der gesamte Bildschirm! Das ist natürlich nicht gerade nützlich. Abhilfe schafft folgendes Programm:

```
LIBRARY "graphics.library"
RastPort&=WINDOW(8)
Flood& RastPort&,Modus,XPos%,YPos%
```

(Wird bei »Modus« eine 1 angegeben, so arbeitet die Anweisung »Flood&« wie in Zeichenprogrammen, wird eine 0 angegeben, wie »PAINT«)
Der Outline-Modus muß jedoch erst definiert werden:

```
POKE RastPort&+27,Randfarbe
FlagsAdr&=RastPort&+32:Flags=PEEKW(FlagsAdr&)
POKEW FlagsAdr&,(Flags OR 8) :REM Hier wird OUTL.aktiviert
```

Ausschalten kann man den Outline-Modus so:

```
FlagsAdr&=RastPort&+32:POKEW FlagsAdr&,0
```

(Hans Holger Rutz/jb)

Menüauswahl

Viele Benutzer haben auf ihren Disketten mehrere Anwendungsprogramme gespeichert. Dabei stellt sich die Frage, wie man das gewünschte Programm am besten und schnellsten startet. Es gibt verschiedene Möglichkeiten: Zum einen startet man von der Benutzeroberfläche Intuition, wobei das Programm ein Icon besitzen muß, was Speicherplatz kostet, zum anderen aus dem CLI, wobei man den korrekten Dateinamen und die Syntax kennen muß. Es gibt aber noch eine Möglichkeit. Man kann sich ein Auswahlmenü erstellen und das gewünschte Programm durch einen Tastendruck starten. Dazu benötigt man ein Programm, mit dem man die im Verzeichnis »Devs:Keymaps« gespeicherte Tastaturbelegung ändern kann. Dazu gibt es verschiedene Programme auf Public-Domain-Disketten. Zuerst überlegt man sich, welche Programme zur Auswahl stehen sollen. Dazu ein Beispiel:
Wir wollen uns eine Menüauswahl für unsere Workbench erstellen. Auf der Diskette befinden sich folgende Programme, die wir in unser Menü aufnehmen wollen:

»FACC II« (Diskbeschleuniger)
»Cruncher« (Crunch-Programm)
»Mach II« (Uhr, Funktionstastenbelegung Sunmouse und so weiter)
»ED« (Texteditor)
»Calc« (Taschenrechner)
»Spooler« (Druckerspooler)
»VirusX« (Antivirus-Programm)

Außerdem brauchen wir noch die beiden Menüpunkte »Workbench« (Laden der Benutzeroberfläche) und »CLI« (Öffnen der CLI-Arbeitsfenster).
Zuerst erstellen wir uns eine Startup-Sequence, welche die einzelnen Menüpunkte zur Auswahl stellt. Diese kann folgendermaßen aussehen:

```
Guardian >NIL: ;Virusüberwachung beim Booten (Public-Domain)
Setmap menu    ;unsere selbsterstellte Keymap aufrufen
Dmouse >NIL:   ;Mouse-Beschleuniger, Screenblanker (Public-
                Domain)
Echo "*e[0 p"  ;CLI-Cursor ausschalten
Berserk        ;Anti-Virus-Programm (Public-Domain)
Border off     ;CLI-Fenster auf maximale Größe und Border
                löschen (Public-Domain)
Setclock >NIL: ;Echtzeituhr laden
LOAD
Setpatch >NIL: ;Systempatch aufrufen
Setfont >NIL:  ;eigenen Font laden
pearl 8
Mount vd0:     ;ASDG-RAM-DISK anmelden
Assign >NIL: t: ;Temporäres Verzeichnis in RAM-Disk
ram:
Assign >NIL: x: ;Diese Zuweisung ist wichtig für unsere Menü-
c:execute       auswahl
Path >NIL: sys: ;Suchpfade
ram: vd0: add
Conman >NIL: -c ;CLI-Shell starten (Public Domain)
                ;Hier kommt die Menüauswahl !!!
```

(Der Inhalt der beiden Echoanweisungen wurde nur der Übersicht halber untereinander geschrieben und muß in der Startup-Sequence jeweils in eine Zeile geschrieben werden.)

```
ECHO "*e[0 p*e[5;28H*e[3;33;42mA500 Workbench 1.2/1.3*e[0m
      *e[8;28HF1.....Load Workbench/EXIT
      *e[9;28HF2.....Open CLI/EXIT
      *e[11;28HF3.....FACC II
      *e[12;28HF4.....MACH II
      *e[13;28HF5.....Printer Spooler"
```

```
Echo "*e[14;28HF6.....Calculator
      *e[15;28HF7.....Cruncher
      *e[16;28HF8.....VirusX
      *e[17;28HF9.....Editor
      *e[18;28HESC....REBOOT*e[30m"
```

Nun wollen wir uns die wichtigsten Teile dieser Startup-Sequence einmal ansehen. Der erste Echobefehl schaltet den Cursor im CLI aus. Das anschließende »Border Off« (Public Domain) macht aus dem CLI-Fenster ein Borderless-Window, so daß man nun den völlig leeren Bildschirm sieht (auch der Mauszeiger ist verschwunden, da Dmouse diesen nach 3 Sekunden Nichtbenutzung ausschaltet). Danach werden die diversen Befehle zum Hochfahren des Systems ausgeführt. Welche Befehle hier stehen, bleibt dem Benutzer überlassen. Durch die Anweisung »Assign x: c:execute« sparen wir uns beim Erstellen der Keymap einiges an Tipparbeit. Ganz am Ende steht die Menüauswahl in den beiden Echoanweisungen. Die Zeichenfolgen »*e[....« (ESC) sind Steueranweisungen für den CLI (vergleiche AMIGA DOS 3'90, Seite 94). Dabei bewirkt die Anweisung »*e[30m«, daß die Vordergrundfarbe gleich der Hintergrundfarbe ist, so daß man bei der Auswahl eines Menüpunktes die Befehlsfolge, die wir dadurch aufrufen, nicht sehen kann, was einen professionelleren Eindruck macht. Wenn wir nun mit dieser Startup-Sequence booten, sehen wir (wenn keine Fehler auftreten) bereits, wie die Menüauswahl aussieht. (Achtung: »Setmap Menu« mit Semikolon (;) am Anfang der Zeile ausklammern, da wir ja noch keine Tastaturbelegung mit dem Namen »Menu« erstellt haben, sonst erhalten wir eine Fehlermeldung.) Nun wollen wir uns dem Aufrufen der Programme zuwenden. Dazu laden wir einen Keymap-Editor und mit diesem unsere normale Tastaturbelegung (normalerweise: »Devs:Keymaps/d«). Dort ändern wir die Belegung der gewünschten Tasten, in unserem Fall die Funktionstasten. Jetzt gibt es zweierlei Möglichkeiten: Bei Programmen, die keine weiteren vorbereitenden Befehle benötigen, reicht es, den Programmnamen einzugeben (zum Beispiel »Cruncher«). Will man das Programm als Hintergrund-Task starten oder benötigt der Menüpunkt mehrere Befehle, so sollte man für jeden Punkt eine Batch-Datei (wie die Startup-Sequence) schreiben. Um diese Batch-Datei aufrufen zu können, müssen wir die Tastatur folgendermaßen belegen:

Menüpunkt	Batch-Datei	Taste	Eintrag in Keymap
Load Workbench	BstartupWB	F1	x: s:startupWB (0d)
Open CLI	startupCLI	F2	x: s:startupCLI (0d)
FACC II	startupFACC	F3	x: s:startupFACC (0d)
MACH II	startupMACH	F4	x: s:startupMACH (0d)
PrinterSpooler	startupSPOOL	F5	x: s:startupSPOOL (0d)
Calculator	startupCALC	F6	x: s:startupCALC (0d)
Cruncher	startupCRUN	F7	x: s:StartupCRUN (0d)
VirusX	startupVIRUS	F8	x: s:startupVIRUS (0d)
Editor	startupEd	F9	x: s:startupEd (0d)
ReBoot	startupBOOT	ESC	x: s:startupBOOT (0d)

Der Eintrag in die Keymap muß ein abschließendes »Return (Hex-Code »0d«) beinhalten, damit die Befehlszeile auch gleich ausgeführt wird. Diese Keymap speichern wir nun unter dem Namen »MENU« im Verzeichnis »DEVS: Keymaps«.
Nun müssen wir nur noch die einzelnen Batch-Dateien erstellen und mit dem Namen speichern, den wir beim Eintrag in die Keymap festgelegt haben, zum Beispiel die Batch-Datei »StartupWB«:

```
setmap d ;dies ist nötig, um unsere Funktionstasten-
                 ;belegung rückgängig zu machen.
copy sys:asdgdisk.info to vd0:Disk.info
copy sys:ramdisk.info to ram:Disk.info
                 ;(Vgl. AMIGA-DOS 3'90, Seite 95)
loadwb ;Benutzeroberfläche laden
echo "*e[0m" ;Farbauswahl im CLI rückgängig machen
endcli ;Menüauswahl beenden
```

Oder die Batch-Datei »StartupCLI«:

```
Setmap d
Newcli CON:0/0/640/256/CONMAN-SHELL FROM sys:s/border
wait 4
Newcli CON:0/0/640/256/CONMAN-SHELL FROM sys:s/border
wait 4
echo "*e[0m"
endcli
```

Dabei bewirkt die Syntax »Newcli .... From sys:s/border«, daß bei Ausführung automatisch die Batch-Datei »sys:s/border« abgearbeitet wird, welche die Befehle »Border off« und »Info« enthält. Deshalb ist auch der Befehl »Wait 4« nötig, um ein "Sägen" des Laufwerks zu vermeiden. Natürlich kann die From-Option auch weggelassen werden.

Nun noch ein Beispiel für die Menüpunkte, die nur ein Programm aufrufen, nämlich die Batch-Datei »startupFACC«.

```
runback >NIL: -3 C:FaccII
```

Bei dieser Syntax wird das Programm als Hintergrund-Task gestartet, und die Menüauswahl bleibt dadurch aktiv. Das heißt, daß weitere Menüpunkte aufgerufen werden können. Die Ausgabe von Runback »started background task« wird an das logische Gerät »NIL:« gesendet, so daß sie nicht auf dem Bildschirm erscheint.

Beim Aufruf von Spooler ist noch ein vorbereitender Befehl nötig, was folgendermaßen aussieht (Batch-Datei »startup-SPOOL«):

```
assign SPOOL: ram:
runback >NIL: c:Spooler
```

So oder ähnlich können wir nun alle benötigten Batch-Dateien schreiben und im Verzeichnis »Sys:s« speichern. Übrigens, der Punkt »REBOOT« startet ein Programm, das mir auf einer Public-Domain-Diskette aufgefallen ist und das System ohne "Affengriff" zurücksetzt. Damit ist alles erledigt, und wir können mit der neuen Workbench booten und alle Menüpunkte einer Prüfung unterziehen. Vorher müssen wir das eventuell eingesetzte Semikolon (;) vor »Setmap Menu« in der Startup-Sequence wieder entfernen.

Nach meinen Erfahrungen ist dieses System der Auswahl recht praktisch, weil die eigentliche Startup-Sequence nur Befehle enthält, die nötig sind. Dabei könnte beim oberen Beispiel sicher der eine oder andere Befehl wegfallen oder durch die schnelleren Befehle aus dem ARP-Projekt ersetzt werden. (Public Domain: verschiedene Befehle, die meist schneller und kürzer sind und eine größere Auswahl von Optionen bereitstellen als die original Amiga-DOS-Befehle). Alle anderen Befehle werden erst dann ausgeführt, wenn sie wirklich gebraucht werden. (Wozu brauche ich die Benutzeroberfläche oder ein CLI-Fenster mit Shell und die RAM-Disks oder die Echtzeituhr, wenn ich nur auf einer Diskette schnell mal die Startup-Sequence mit dem Editor ändern will?) Dadurch ist das System schneller hochgefahren und läßt sich jedesmal optimal an die gerade aktuellen Bedürfnisse anpassen.

(Norbert Dimpfl/jb)

Anmerkung der Redaktion: *Obwohl in diesem Tip sehr viel mit PD-Programmen gearbeitet wird, haben wir ihn abgedruckt. Er ist durch leichte Abänderungen auch auf "normale" Workbench-Programme anwendbar. Außerdem sind die genannten PD-Programme preiswert bei vielen PD-Händlern zu bekommen.*

Michael Anton

Das Amiga-DOS – leicht und verständlich!

Im CLI wohnlich eingerichtet...

In dieser Folge des CLI-Kurses wollen wir uns diesen ordentlich einrichten und eine praktikable Startdiskette dafür erstellen. Dabei wird es sich nicht vermeiden lassen, daß Sie auch mit anderen DOS-Befehlen konfrontiert werden.

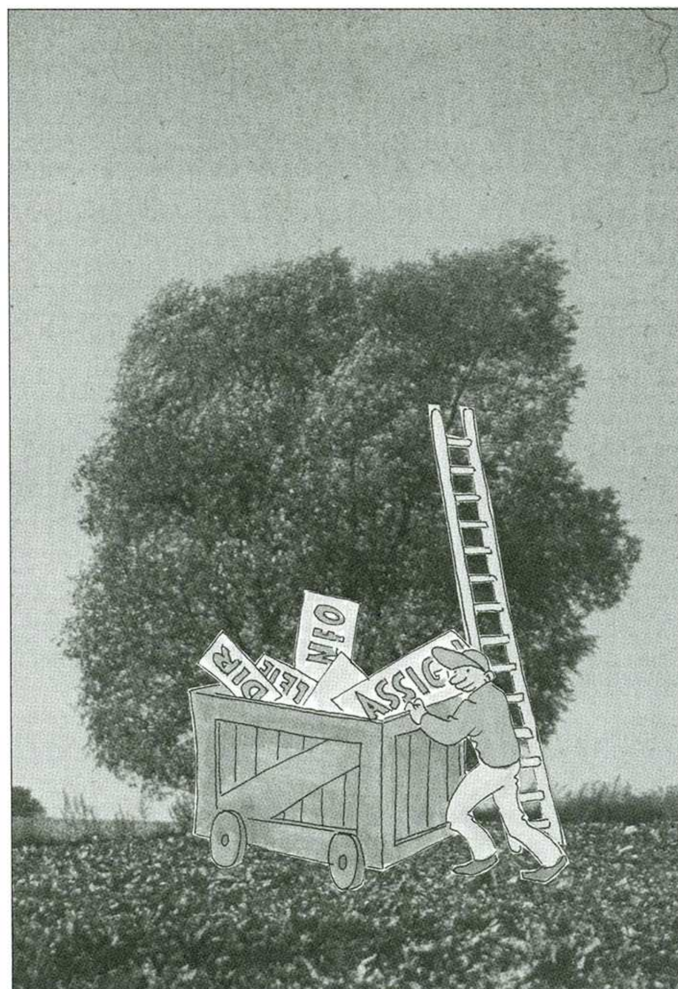
Zuerst eine kleine Entschuldigung an alle, die noch immer an der Aufgabe aus der letzten Folge herumknobeln. Mit den Vorarbeiten haben wir Sie tatsächlich etwas an der Nase herumgeführt. Sicher geht es, wenn die benötigten Dateien und Verzeichnisse von Hand auf die Startdiskette übertragen werden. Wer's gemacht hat, hat sicher auch gemerkt, daß es eine recht langwierige Sache ist, bei der man leicht das eine oder andere übersehen kann. Viel einfacher wäre es, die normale Workbench-Diskette einfach zu duplizieren, entweder über die Workbench oder über DISKCOPY – das geht nicht nur schneller, sondern überträgt auch wirklich alle Dateien an die richtige Stelle.

Aber eines haben wir damit schon gelernt: Bei der Arbeit mit dem CLI muß mitgedacht werden, wenn die Arbeit erfolgreich und ökonomisch sein soll. Jede Operation erfordert einiges an Tipparbeit, Fehler machen Mehrarbeit nötig. Also sollte man sich jede Aktion gut überlegen und Alternativen und Erleichterungen in Erwägung ziehen. Wenn Sie beispielsweise beabsichtigen, in einer Sitzung besonders häufig mit Dateien in einem Unterverzeichnis zu arbeiten, sollten Sie auf dieses Verzeichnis einen Verweis mit ASSIGN definieren. Ein einfaches

ASSIGN WORK: DF1:TEXTE/BRIEFE/1990

macht nur einmal das Tippen des Monster-Pfades notwendig, in Zukunft kann man die dortigen Dateien wesentlich schneller durch die Angabe des logischen Pfades erreichen. Ähnliches gilt auch für die generelle Organisation von Verzeichnisbäumen oder den Wechsel innerhalb solcher Bäume. Aber nun sind Sie gewarnt, mit etwas Praxis werden Sie sehr schnell den richtigen Dreh herausfinden...

Kursfahrplan
Teil 1 – Einführung in den CLI
Teil 2 – Was kann Amiga-DOS
Teil 3 – Stapeldateien
Teil 4 – Neu dabei bei 1.3
Teil 5 – Alternativen zum CLI



Startdiskette, die Zweite

Kommen wir zum Problem der Startdiskette zurück. Wenn Sie von Hand kopiert haben, werden Sie nach dem Booten mit der Diskette vielleicht an einer oder anderer Stelle doch Fehlermeldungen bekommen haben. Das liegt daran, daß das DOS auf bestimmte Verzeichnisse besteht und auf deren Fehlen ungnädig reagiert. Diese (unter der Workbench unsichtbaren) Verzeichnisse werden normalerweise beim Booten definiert, geschieht dies nicht, greift das DOS ins Leere. Diese Verzeichnisse sind L:, LIBS: und DEVS:, in denen wichtige Systemdateien wie Gerätetreiber und Libraries stehen, FONTS: für Zeichensätze, C: für die DOS-Befehle und T: zur Ablage von temporären Dateien. Weiterhin wichtig ist S:, in dem Voreinstellungen stehen – und dies ist genau der Knackpunkt, aber dazu später mehr.

Da es also wichtig ist, das DOS nicht allzusehr zu vergrätzen, fangen wir nochmals mit der Erstellung der Startdiskette an, diesmal richtig – also mit einer 1:1-Kopie der Workbench-Diskette, wahlweise mit DISKCOPY oder der Workbench! Legen Sie nun diese Kopie in DF0:, und lösen Sie einen Reset aus. Sie werden mit Recht einwenden, daß dieser Vorgang nur wieder zur Workbench führt. Sie haben recht – aber auch nicht. Der Amiga nudelt eine ganze Weile auf der Diskette herum, also spielt sich irgend etwas im Rechner ab (nähere Erklärung folgt weiter unten). Da dies einige Zeit dauert, können wir uns in Ruhe auf die nächste Aktion vorbereiten: das Drücken einer Taste, und zwar der Tastenkombination CTRL+D. Wenn Sie nicht zu lange gezögert haben, erscheint statt der Workbench eine Eingabezeile, die an den CLI erinnert – und das ist es auch fast schon. Zwar nicht das gewohnte Fenster, welches unter der Workbench geöffnet wird, aber das gleiche Prinzip.

Die Startup-Sequenz wird beim Booten automatisch abgearbeitet

Offensichtlich laufen beim Booten Vorgänge automatisch ab, die wir durch das Drücken von CTRL+D unterbrochen haben. Solche Unterbrechungen – auch Interrupts genannt – werden durch gleichzeitiges Drücken von CTRL und einer anderen Taste erreicht. Es sind vier Interrupts dieser Art vorhanden: CTRL+C bricht ein gerade laufendes Programm ab – so-

fern dieses den Interrupt abfragt, CTRL+D bricht Stapelverarbeitungen ab, CTRL+E und CTRL+F sind für besondere Fälle vorgesehen. Aber was ist nun wieder 'Stapelverarbeitung'? Ganz einfach: wieder ein Relikt aus der Steinzeit der EDV. 'Früher' bestanden Computerprogramme aus einem Stapel Lochkarten, die vom Computer verarbeitet wurden, daher der Name, auf englisch auch als Batch-Job bekannt. Moderne Rechner machen es ähnlich, allerdings werden hier keine Lochkarten mehr verfüttert, sondern Dateien erstellt. Diese Stapel- oder Batch-Dateien enthalten in Form einer Textdatei eine Folge von Anweisungen, die der Computer nacheinander abarbeitet. Dazu muß nur ein Programm gestartet werden – eben der Batch-Job; der Rest geht von allein.

Nichts anderes geschieht in unserem Fall. Im Verzeichnis S: der Startdiskette steht eine solche Datei, die beim Booten automatisch abgearbeitet wird. Sie nennt sich STARTUP-SEQUENCE und enthält eine Folge von Befehlen, die letztlich zur Anmeldung der Systemverzeichnisse und zum Start der Workbench führen.

Steuersequenzen für Bildschirmausgabe

Farbe und Form der Bildschirmausgabe lassen sich auf dem Amiga in weiten Grenzen variieren, dies kommt im CLI besonders bei den Befehlen ECHO und PROMPT zum Tragen. Eingeleitet werden diese Steuersequenzen durch das Escape-Symbol, auf dem Amiga durch '*e' dargestellt.

Sequenz	Funktion
*ec	*Bildschirm löschen, Normaldarstellung*
*e[0m	*Normaldarstellung*
*e[1m	*Fettschrift ein*
*e[2m	*Textfarbe 2 (Schwarz)*
*e[3m	*Kursivschrift ein*
*e[30m	*Textfarbe 0 (Blau)*
*e[31m	*Textfarbe 1 (Weiß)*
*e[32m	*Textfarbe 2 (Schwarz)*
*e[33m	*Textfarbe 3 (Rot)*
*e[4m	*Unterstreichen ein*
*e[40	*Hintergrundfarbe 0 (Blau)*
*e[41	*Hintergrundfarbe 1(Weiß)*
*e[42	*Hintergrundfarbe 2 (Schwarz)*
*e[43	*Hintergrundfarbe 4 (Rot)*
*e[7m	*Invers ein (Farbe 0 auf Farbe 1)*
*e[8m	*Unsichtbare Schrift (Farbe 0 auf Farbe 0)*

Über ähnliche Sequenzen läßt sich auch die Cursorposition steuern:

*e[nH	*Position in Zeile n*
#J	*Löscht Bildschirm ab aktueller Position*
#nL	*Verschiebt Bildschirm um n Zeilen nach unten*
#nM	*Wie oben, mit Löschen der aktuellen Zeile*
#S	*Verschiebt CLI-Inhalt n Zeilen nach oben*
#T	*Verschiebt CLI-Inhalt n Zeilen nach unten*

Zur Verwendung dieser Sequenzen müssen sie in Hochkommata eingeschlossen werden, sofern sie mit einem Text mit Leerzeichen verwendet werden. Die Attribute bleiben so lange bestehen, bis sie geändert oder ausgeschaltet werden. Falls die reservierten Steuerzeichen auf dem Bildschirm dargestellt werden sollen, ist für '*' '**' und für '“' '*“' einzugeben. '*n' bewirkt einen Zeilenvorschub.

Mit diesen Sequenzen können Ausgaben in Stapeldateien via ECHO ansprechend gestaltet werden. Eine andere Verwendung ist die individuelle Gestaltung der Eingabeaufforderung – auch Prompt genannt. Normalerweise zeigt DOS im CLI nur die Nummer der aktuellen Task an, diese Anzeige kann jedoch durch Texte und Steuersequenzen variiert werden. Wird die Anzeige der Task-Nummer gewünscht, muß dies an entsprechender Stelle durch '%n' markiert werden.

Nach der Eingabe von PROMPT “*e[33m*e[3mSie wünschen? >*e[0m“ fragt der Amiga kursiv und in Rot nach den Wünschen des Anwenders, um sie in normaler weißer Darstellung entgegenzunehmen. (Der Standard-Prompt entspricht in etwa PROMPT “%n>“.

(Die Angaben der Farben entsprechen der Grundeinstellung, über PREFERENCES können sie jederzeit verändert werden!)

Tabelle. Übersicht über die Sequenzen zur Ansteuerung der Bildschirmausgabe

Was machen wir nun aber mit der Startdiskette? Sowohl die Unterbrechung der Sequenz beim Booten als auch der völlige Verzicht auf sie sind unbefriedigend, aber jedesmal erst unter der Workbench ein CLI-Fenster zu öffnen, ist für eine Startdiskette auch nicht der ideale Weg. Aber da es sich um eine Textdatei handelt, könnte man sie ja mit einfachen Mitteln den Gegebenheiten anpassen.

Großputz auf der Startdiskette

Vor den Erfolg haben die Götter jedoch die Schweißfüße gesetzt. Normalerweise wird es auf der Startdiskette recht eng mit dem Platz. Für uns lautet daher die Devise: Weg mit allem, was nicht nötig ist. Aber was ist nicht lebenswichtig? Eine gute Frage, die vor allem auch von der Rechnerkonfiguration abhängt. Einige Dateien der Diskette werden nur benötigt, wenn eine entsprechende Hardware eingebaut oder angeschlossen ist, beispielsweise Festplatte oder PC-Karte. Zum anderen kann man sich aber auch getrost von einigen Programmen trennen. CLOCK und NOTEPAD sind meines Erachtens ideale Kandidaten für den letzten Fall, wer nur selten mit den PREFERENCES spielt, kann auch diese von der Scheibe verbannen. Wer keine PC-Karte sein eigen nennt, kann noch mehr Platz schaffen: Das ganze SIDECAR-Verzeichnis kann verschwinden, später auch das PC-Verzeichnis, letzteres allerdings erst nach einigen Vorarbeiten! Auch EXPANSION ist potentieller Kandidat, ab Werk steht hier normalerweise nur eine Library für die PC-Karte, also möglichst weg damit.

Startup-Sequenz für Individualisten

Ziel der Löschaktion ist nicht nur, Platz für die im folgenden beschriebenen Editieraktionen zu machen, der Platz kann auch gut für andere Zwecke verwendet werden, beispielsweise um einen Virenkiller wie VirusX oder andere Hilfsmittel auf der Startdiskette unterzubringen. Zur Anzeige des freien Platzes auf den Datenträgern kann INFO verwendet werden. Es zeigt nach dem Aufruf im CLI an, welche physikalischen Laufwerke wie belegt sind und welche Disketten das System 'kennt' (Bild 1). Auf der Startdiskette sollten zum Editieren etwas mehr als 3% frei sein, mehr ist auch nicht zu verachten. Ist der Platz durch Einsatz von DELETE vorhanden, kann es losgehen.

Wechseln wir also in das Verzeichnis S: und schauen uns die Startup-Sequenz genauer an. Dazu dient ein neues DOS-Kommando, nämlich TYPE. Mit ihm können Dateien auf dem Bildschirm angezeigt, aber auch nach bestimmter Bearbeitung an Ausgabegeräte oder in andere Dateien gesandt werden. Die einfachste Form des Aufrufes ist

```
TYPE dateiname
```

was zur Ausgabe der Datei auf den Bildschirm führt. Wird ein zweiter Dateiname angegeben, wird der Inhalt der ersten Datei in die zweite Datei kopiert, gibt man statt dessen ein Gerät an, erfolgt die Ausgabe der Datei dorthin. Die Richtung kann(!) zusätzlich mit den Schlüsselwörtern FROM und TO angegeben werden. So kopiert

```
TYPE FROM Text TO Kopie
```

den Inhalt in eine andere Datei, während

```
TYPE Text TO PRT:
```

eine Ausgabe auf dem Drucker bewirkt. Mit der Option 'H' erfolgt die Darstellung im Hex-Modus, bei der Option 'N' werden die Zeilen numeriert. Mit

```
TYPE Text TO PRT: OPT H
```

wird beispielsweise ein numerierter Ausdruck von 'Text' erzeugt.

Für unseren Fall reicht jedoch die einfache Anzeige mit

```
TYPE Startup-Sequence
```

um den Inhalt am Bildschirm zu sehen. Durch Drücken einer beliebigen Taste außer 'Backspace' und 'Return' kann die Anzeige angehalten werden, nach Drücken der beiden anderen Tasten wird die Ausgabe fortgesetzt.

Mit dem Anschauen ist es aber nicht getan, um den Inhalt der Datei zu ändern, muß sie editiert werden. Dafür stellt das DOS einen Editor zur Verfügung, mit dem Texte

bequem bearbeitet werden können. Der Helfer heißt ED und wird normalerweise mit

```
ED dateiname
```

aufgerufen, mit 'SIZE xxx' kann die auf 40 kByte begrenzte Textlänge erweitert werden. Für unsere Arbeit reicht jedoch der normale Aufruf, um diese Datei bequem auf dem Bildschirm zu betrachten. Sie sieht in etwa so aus, wie in Bild 2 dargestellt. Mit den Cursortasten können wir uns nun im Text bewegen und Änderungen vornehmen. Es stehen viele Funktionen zur Verfügung, wir benötigen im wesentlichen jedoch nur eine Kombination: CTRL + B zum Löschen von Zeilen. Die übrigen Funktionen können Sie Ihrem Handbuch entnehmen, ED ist im Gegensatz zu manchen anderen Dingen dort recht ausführlich beschrieben. Nachdem Sie die Datei derart abgeändert haben, daß sie wie Bild 3 aussieht, drücken Sie nacheinander 'Esc', 'X' und 'Return', um die Änderungen zu speichern. Booten Sie nun den Rechner erneut, und alles sieht gleich ganz anders aus.
Das Booten geht nicht nur schneller vonstatten, statt der Workbench meldet sich der Amiga nun auch mit einem CLI-Fenster, welches nach der Verwendung der Tastatur schreit.
Der Aufbau des Beispiels gilt nur für die Version 1.2, bei der Version 1.3 der Workbench wurde die Startup-Sequenz aufgeteilt, das Vorgehen ist jedoch weitgehend identisch. Die neue Version der Startup-Sequenz kann jedenfalls ohne Probleme übernommen werden.

Was haben wir eigentlich gemacht?

Es kann natürlich nicht der Sinn eines Kurses sein, Sie kommentarlos von einem Listing zum anderen zu schicken. Also nehmen wir uns die beiden Versionen der Startup-Sequenz Zeile für Zeile vor und schauen, was passiert ist oder passiert wäre. Einige der enthaltenen Befehle sind neu, manche von ihnen können jedoch erst in der nächsten Folge besprochen werden.
Das erste Listing beginnt mit dem Befehl BINDDRIVERS. Mit ihm werden Treiberprogramme eingebunden, die im

```
1.SYS:> info

Mounted disks:
Unit      Size    Used    Free Full Errs   Status    Name
DF0:      837K     974     784  55%    0  Read Only  AmigaDOS7.90
DF1:      837K     206    1552  11%    0  Read Only  Emp
DF2:      No disk present
DH0:       47M   28855   67083  30%    0  Read/Write Fetzplatte
RAM:        1K       4       0 100%    0  Read/Write RAM DISK

Volumes available:
AmigaDOS7.90 [Mounted]
Emp [Mounted]
Fetzplatte [Mounted]
RAM DISK [Mounted]
1.SYS:>
```

Bild 1. Mit INFO lassen sich Informationen über die Kapazitäten der Laufwerke anzeigen

Verzeichnis EXPANSION stehen. Ist keine Hardware vorhanden, die solche Treiber benötigt, kann dieser Befehl entfallen. Mit ECHO werden Meldungen auf dem Bildschirm ausgegeben, in unserem Fall ist es nur ein einfacher Text. Steuerzeichen können für Abwechslung in der Ausgabe sorgen, näheres entnehmen Sie der Tabelle. Den folgenden ASSIGN-Befehl kennen Sie bereits aus der ersten Folge des Kurses, die ihm folgenden Blöcke mit IF, ELSE, ENDIF und EXISTS dienen der Einbindung von Festplatten. Die Funktion dieser Befehle soll jedoch erst in der nächsten Folge beschrieben werden. Ihre generelle Funktion ist jedoch der bedingte Ablauf der Sequenz. WAIT hält die Abarbeitung für eine oder bis zu einer bestimmten Zeit an (hierzu an anderer Stelle mehr), mit DJMOUNT wird eine Festplatte eingebunden.
Mit ADDBUFFERS wird einem bestimmten Laufwerk (hier dem Bootlaufwerk) eine bestimmte Anzahl von Blöcken zu 512 Byte als Puffer zugewiesen. Dies führt zu einer schnelleren Arbeit des entsprechenden Laufwerks, geht allerdings auf Kosten des Arbeitsspeichers. Werte von mehr als 30 (= 15 kByte) sind jedoch selten sinnvoll. Die folgenden Kombinationen aus Bedingungen und Zuweisungen dienen der Festlegung der nötigen Systemverzeichnisse, wobei die Bootdiskette als Bezug gewählt wird. Es wird jeweils geprüft, ob auf der Bootdiskette ein entsprechendes Verzeichnis vorhanden ist, und dieses dann zum Systemverzeichnis deklariert. Am Ende dieser Kette kommt ein neuer Befehl ins Spiel, der für ein bequemes Arbeiten sehr wichtig ist. Normalerweise sucht DOS nur im logischen C:-Verzeichnis nach Programmen. Mit PATH wird dieser Bereich jedoch erweitert, so daß beim Aufruf eines Programms nicht nur C:, sondern auch andere Verzeichnisse durchsucht werden. Die Verwendung von PATH ist sehr sinnvoll: Bei der Verwendung von Disketten und RAM-Disk können wichtige Kommandos aus C: nach RAM: verlagert werden und sind dort schneller verfügbar, selbst wenn die Diskette von C: nicht verfügbar ist. Auf Festplatten ist die Verwendung von PATH sowieso sinnvoll. Man kann zwar häufig benötigte Tools nach C: kopieren, verliert dann aber sehr schnell die Übersicht, da C: nur DOS-Befehle enthalten sollte. Statt dessen können solche Tools in Verzeichnissen wie TOOLS oder UTIL zusammengefaßt und mit PATH allgemein zugänglich gemacht werden. Die Syntax von PATH ist

```
Ed 1.14
BindDrivers
echo "A2000  Workbench 1.2 D  V33.56  2-JUN-87*N"
Assign WB: sys:
IF EXISTS sys:s/JH0
   echo "Initializing Janus ...*N"
   Wait 25
   Echo "Mounting Janus Hard Disk*N"
   DJMount
   IF EXISTS JH0:
      Assign WB: JH0:
      Echo "Transferring control to JH0: *N"
   ELSE
      Assign WB: sys:
   EndIF
EndIF
IF EXISTS sys:s/DH0
   Assign WB: DH0:
   Echo "Transferring control to DH0: *N"
EndIF
AddBuffers WB: 30
CD WB:
IF EXISTS WB:c
   Assign c: WB:c
EndIF
IF EXISTS WB:s
   Assign s: WB:s
EndIF
IF EXISTS WB:t
   Assign t: WB:t
```

Bild 2. So sieht die von Commodore erstellte Startup-Sequenz in etwa aus (in der Version 1.2 für den A2000)

```
PATH ADD pfadname
```

Ein Aufruf ohne Parameter zeigt die aktuellen Pfade.

Sag' mir, wo die Daten sind...

Nach diesen Zuweisungen folgt die Anzeige des Inhalts der RAM-Disk und die Festlegung eines Suchpfades dorthin. Die Anzeige ist insofern nötig, da das Icon für die RAM-Disk nur dann auf der Workbench erscheint, wenn bereits ein Zugriff auf RAM: erfolgt ist. (Als Gerät ist die RAM-Disk jedoch stets verfügbar, so daß der Aufruf nicht unbedingt sein muß.) Mit FAILAT kann die Verarbeitung von Stapeldateien abgebrochen werden, aber dazu mehr in der nächsten Folge. SETCLOCK ist ebenfalls ein neuer Befehl, er dient zur Arbeit mit der gepufferten Systemuhr – sofern eine solche vorhanden ist. OPT LOAD lädt den Inhalt der Hardware-Uhr in die Software-Uhr, OPT SAVE funktioniert in umgekehrter Richtung. Die erstmalige Einstellung von Uhrzeit und Datum erfolgt unter der Workbench mit PREFERENCES oder mit DATE im CLI. Die Syntax von DATE ist

```
DATE TIME HH:MM:SS DATE TT-MMM-JJ
```

(Bei der Angabe des Datums ist zu beachten, daß die Angabe des Monats in Form von drei Buchstaben erfolgt, wobei amerikanische Konventionen gelten, also 'May' statt 'Mai' und so weiter)
Der Befehl SETMAP dient zur Einstellung der Tastatur auf die landesspezifischen Konventionen, sprich auf Umlaute und Sonderzeichen. Im Verzeichnis DEVS/KEYMAPS befinden sich vorbereitete Tabellen, die Tabelle D (daher auch SETMAP D) ist für den deutschsprachigen Raum am besten geeignet. Und nun kommen wir zu dem Punkt, der uns am meisten berührt. Mit den nächsten beiden Befehlen, LOADWB und ENDCLI (der Block mit der Einbindung der Harddisk sei ignoriert!), wird nichts anderes getan, als die Workbench geladen und der während des Bootens aktive CLI geschlossen. Und spätestens hier können wir ansetzen! Theoretisch reicht es, diese beiden Befehle zu entfernen, um nach dem Booten direkt im CLI zu landen. Das ist aber noch nicht alles...

Die Startup-Sequenz entschlacken

Einige der in der Startup-Sequenz getätigten Aktionen sind überflüssig und verlangsamen die Ausführung erheblich, ist doch für jeden Befehl ein Diskettenzugriff notwendig. Schneller geht es also, wenn nicht benötigte Teile einfach entfernt werden. Wer keine PC-Karte besitzt oder glücklicher Besitzer einer Autoboot-Platte ist, kann auf den Block zwischen ASSIGN WB: SYS: und ADDBUFFERS WB: 30 verzichten, des weiteren auf den Block zwischen Laden der Workbench und Verlassen des CLI. Da bei unserer Erstellung der Startdiskette die nötigen Verzeichnisse mit Sicherheit vorhanden sind, kann bei der Zuweisung der Systemverzeichnisse auch auf die Abfragen verzichtet und nur das ASSIGN durchgeführt werden. Über die Einbindung via BINDDRIVERS und SETCLOCK sowie der Anmeldung von RAM: muß die Ausstattung entscheiden, das Ende der Sequenz ist Geschmackssache.
Im Prinzip reicht es, nach SETMAP D die Sequenz abzubrechen, man ist dann auf alle Fälle in einem vollwertigen CLI. Allerdings ist es dann etwas schwierig, notfalls auf die Workbench zu gelangen.

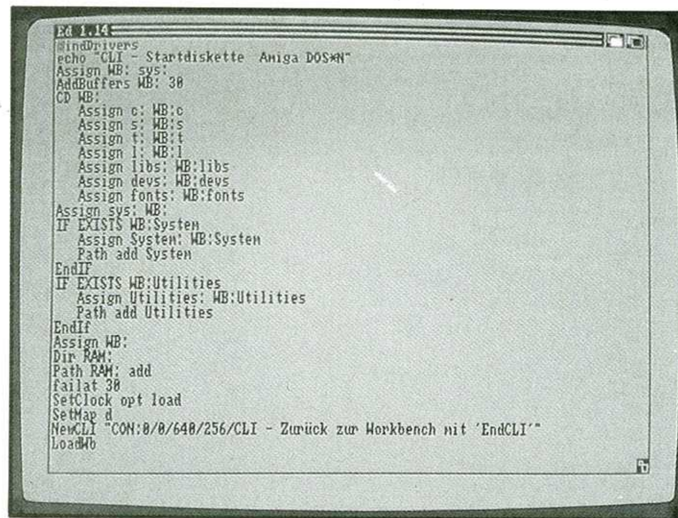


Bild 3. Die neue Variante arbeitet wesentlich schneller und führt direkt in den CLI

Zu diesem Zweck müssen im CLI zunächst LOADWB zum Laden der Workbench und ENDCLI zum Wechsel in die Workbench eingegeben werden. (Ein ENDCLI ohne LOADWB macht die weitere Bedienung etwas problematisch!)
Die vorgestellte Lösung basiert darauf, daß innerhalb der Startup-Sequenz ein neuer CLI geöffnet wird. Der Befehl hierzu lautet NEWCLI. NEWCLI erlaubt die Übergabe von diversen Parametern, wobei die Syntax

```
NEWCLI "CON:x/y/b/h/t"
```

ist. Dabei sind x und y die Koordinaten der linken oberen Ecke des CLI-Fensters in Pixeln, b und h dessen Breite und Höhe, t ist eine Zeichenkette, die den Fenstertitel bildet. (In unserem Beispiel wird ein Fenster auf maximale Größe eröffnet, dessen Titel einen Hinweis zur Bedienung enthält.) Die Ausführung der Sequenz wird dadurch so lange verzögert, bis dieser CLI mit ENDCLI beendet wird.
Anschließend erfolgt automatisch der Start der Workbench. Der Nachteil ist, daß durch die auf Eis gelegte Sequenz etwas Rechenzeit verschenkt wird, im Zuge des Komforts kann man jedoch darauf verzichten – als Alternative kann man jedoch, wie bereits gesagt, nach SETMAP D abbrechen. Innerhalb der Startup-Sequenz lassen sich auch noch weitere Aktionen tätigen, beispielsweise das Kopieren wichtiger Befehle nach RAM: oder der Start zusätzlicher Programme. Dies hängt jedoch von Ihrer Hardware und Ihren Wünschen ab, so daß an dieser Stelle kein Patentrezept für die individuelle Gestaltung gegeben werden kann.

Die Aufgabe

Richtige 'Hausaufgaben' sollen an dieser Stelle nicht aufgegeben werden. Stellen Sie sich statt dessen eine CLI-Diskette nach Ihren Wünschen zusammen, wobei die Erkenntnisse des Kurses eine Hilfe sein sollen. Denken Sie dabei vor allem an die Einbindung zusätzlicher Werkzeuge. Gestalten Sie die Umgebung nach Ihrem Geschmack, die Tabelle kann Ihnen eine Hilfe dabei sein. Und beachten Sie immer: bei der Arbeit mitdenken! Es geht nicht um Mausklickerei, sondern um sinnvolle Aktionen. Lernen Sie den Umgang mit ED, und machen Sie sich mit der Stapelverarbeitung vertraut. Erstellen Sie mit den bisherigen Erkenntnissen Jobs, die Sie für sinnvoll halten. Hierzu noch ein Hinweis: Die Startup-Sequenz ist nicht die einzige Form der Stapelverarbeitung. Eine Textdatei mit solchen Anweisungen kann auch im CLI mit dem Befehl EXECUTE gestartet werden. Ist das ein Angebot?

Besitzer der Version 1.3 haben darüber hinaus noch mit anderen Problemen bei der Anpassung zu kämpfen: Diese Version der Startup-Sequenz greift recht intensiv auf neue Features wie RESIDENT zurück. Hier muß zusätzlich mit dem Speicherbedarf und der Arbeitsgeschwindigkeit jongliert werden, um eine gute Anpassung zu erzielen. (Mehr dazu in einer der nächsten Folgen.)
Noch kurz zurück zum Thema Löschen des Verzeichnisses C:. In der Normalversion der Startup-Sequenz wird gegen Ende ein Pfad auf dieses Verzeichnis definiert. Daher ist es unmöglich, dieses Verzeichnis nach dem Booten mit der Originalsequenz zu löschen, statt dessen erscheint die Fehlermeldung 'Object in use'. Nach dem Booten mit der modifizierten Version ist jedoch auch dieses Verzeichnis problemlos löschbar.

Ein Ausblick auf das, was kommt

Die Arbeit von DOS ist nicht nur auf das Ausführen von Programmen beschränkt, sondern kümmert sich auch stark um die Belange des Anwenders. Stapelverarbeitung ist ein wichtiger Aspekt dieses Themas, der an dieser Stelle noch nicht vollständig behandelt worden ist – ganz im Gegenteil. Was man mit derlei Stapeln von DOS-Befehlen erreichen kann, dürfte Ihnen bis jetzt noch nicht geläufig sein, jedenfalls, wenn Sie noch nicht fest im Sattel sitzen. Vielleicht ein kleiner Vorgeschmack auf das, was kommt:

```
ask "Soll die Workbench angezeigt werden?"
if not warn
New CLI CON:0/0/640/256/MyCLI
endif
endcli>NIL:
```

Werden diese Zeilen an das Ende der Startup-Sequenz eingefügt, kann man hinterher wählen, ob man beim Start in die Workbench oder ins CLI gehen will. Einfach, nicht wahr?
Die nächste Folge wird voll und ganz diesem Thema gewidmet sein. Aber keine Panik: Auch wenn das Thema auf den ersten Blick etwas 'altmodisch' aussieht, hat es bei der Arbeit mit dem CLI doch seine Vorteile. Aber lassen Sie sich überraschen...

(jb)

Andreas Polk

Freie Fahrt für Modula2

Der Kurs für Ein- und Umsteiger

Kursfahrplan 

Teil 1: Der Compiler M2Amiga

Teil 2: Die ersten Befehle

Teil 3: Schleifen und Verschachteln

Teil 4: Prozeduren und Variablen

Teil 5: Weitere Datentypen

Teil 6: Zeiger und Datenstrukturen

Teil 7: Modulkonzept

Teil 8: Systemprogrammierung



Den Amiga programmieren – ein Thema, das viele Amiga-Besitzer immer wieder fasziniert. Doch mit welcher Sprache? Eine noch nicht so weit verbreitete und auch etwas eigenwillige Sprache ist Modula2, die wir Ihnen in diesem Kurs etwas näherbringen wollen.

Wenn Sie bisher nur zu den Anwendern zählten (was im übrigen auch nicht schlimm ist!), so haben Sie sich sicherlich schon einmal gefragt, wie man Programme selbst schreibt. Sie benötigen dazu eine Programmiersprache. Amiga-BASIC wird jedem Amiga umsonst beigelegt. Allerdings ist diese Sprache auch nicht das Gelbe vom Ei.

Interpreter oder Compiler

Um in Modula2 zu programmieren, brauchen Sie einen Compiler. Was sind aber Compiler und Interpreter?

Da der Computer weder BASIC noch Modula2 versteht, muß ein Programm, welches in einer dieser Sprachen geschrieben wurde, in eine für den Computer verständliche Form (Maschinensprache) gebracht werden. Dies kann auf zweierlei Arten geschehen, zum einen mit einem Compiler und zum anderen mit einem Interpreter. Ein Interpreter übersetzt einen Befehl in die für den Computer verständliche Form und führt diesen Befehl dann direkt aus. Danach wird der nächste Befehl übersetzt und ausgeführt, und so weiter, bis das Programm beendet wird. Die Übersetzung findet also erst während des Programmablaufs statt.

Ein Compiler dagegen arbeitet anders. Er übersetzt das Programm mit einem Mal in eine für den Computer verständliche Form und speichert diese Datei dann ab. Der Anwender kann nun diese Datei als Programm starten. Während des Programmablaufs muß nicht mehr der nächste Befehl übersetzt werden, wie dies bei einem Interpreter der Fall ist. Jede dieser Vorgehensweisen hat Ihre Vor- und Nachteile. Der Vorteil bei einem Interpreter ist der, daß die Bedienung meist sehr einfach ist und man leichter mit ihm arbeiten kann. Auch die Wartezeit für das separate Erstellen des für den Computer verständlichen Programms (diesen Vorgang nennt man Compilieren) fällt



Der Autor

Andreas Polk erblickte am 26.1.1971 in Kettwig an der Ruhr das Licht der Welt. Mit dem sechsten Lebensjahr fing wie üblich der Schulalltag an, den er vor kurzem nach insgesamt 13 Jahren mit dem Abitur in der Tasche endlich hinter sich ließ.

Zum Hobby 'Computer' kam er vor vier Jahren über einen geliehenen TI 99/4A. Schon bald wechselte er zum C64, und ein halbes Jahr später folgte ein Amiga 500, dann ein Amiga 2000.

Seine Arbeit als Autor begann er vor dreieinhalb Jahren bei Data Becker in Düsseldorf, wo er erst zuerst als Aushilfe, dann als Buchautor arbeitete. Zur Zeit nennt er vier Bücher sein Werk (drei über den Amiga, darunter 'Die besten Amiga-Utilities', eines über den C64). Im Herbst wird das fünfte folgen.

Seine bevorzugten Gebiete sind Anwendungen, dort vor allem MIDI und Animationen.

Doch nicht nur der Computer ist sein Hobby. Neben Unternehmungen mit der Freundin spielt er auch sehr gerne Tennis und Klavier.

weg. Doch damit wären wir auch schon direkt bei dem größten Nachteil: Da der Computer die einzelnen Befehle während des Programmablaufs in Maschinensprache umsetzt, wird die Ausführungszeit des Programms erheblich verlangsamt, denn zwischen zwei Befehlen muß immer erst ein Befehl übersetzt werden, was natürlich Zeit kostet. Dies wird mit dem Compiler umgangen. Er wandelt erst das gesamte Programm in Maschinensprache um. Startet man dieses Programm, so sind alle Befehle direkt ausführbar. Es muß nichts mehr übersetzt werden. Außerdem ist ein einmal übersetztes Programm unabhängig von seinem Compiler lauffähig, während ein mit einem Interpreter erstelltes Programm nicht ohne den Interpreter lauffähig funktioniert.

Alles bezieht sich in diesem Kurs auf den Modula2-Compiler der Firma A. + L. Meier-Vogt mit dem Namen M2Amiga. Er ist der wohl umfangreichste für den Amiga erhältliche Compiler. Außerdem ist er am meisten verbreitet, und es gibt eine ganze Public-Domain-Reihe zu diesem Compiler. Daß er ganz in Deutsch gehalten ist (er wurde in der Schweiz programmiert), spricht ebenfalls für ihn. Ein weiterer Grund, warum wir ausgerechnet diesen Compiler besprechen, liegt darin, daß A. + L. jedem Interessierten die Möglichkeit bietet, die Sprache für wenig Geld kennenzulernen. Sie finden nämlich eine abgespeckte Version von M2Amiga auf der Fish-Disk 113. Mit dieser Demoversion können Sie schon eigene Programme erstellen und den Compiler erst einmal in Ruhe begutachten, bevor Sie sich entscheiden. Alle in diesem Kurs vorgestellten Programme werden mit dieser PD-Version compilierbar sein, es sei denn, es wurde ausdrücklich etwas anderes gesagt.

Warum Modula2?

Diese Frage ist sicherlich berechtigt, weshalb ich kurz darauf eingehen möchte. Man kann nicht pauschal sagen, daß eine Sprache die beste ist. Jede Sprache hat ihren speziellen Anwendungsbereich und somit ganz bestimmte Schwerpunkte. Die Sprache 'C' beispielsweise läßt dem Programmierer sehr viele Freiheiten. Manche Befehle sind sehr komplex. Dies hat den Vorteil, daß C-Programme recht kompakt zu programmieren sind. Der Nachteil liegt jedoch darin, daß der Programmcode, im folgenden 'Source-Code' (Quell-Code) genannt, sehr leicht unübersichtlich wird. Gerade dies macht C bei einigen Leuten verhaßt, da Sie den Einstieg nicht geschafft haben. Die ihn geschafft haben, befürworten jedoch gerade diese Komplexität. BASIC hat den Vorteil, sehr leicht erlernbar zu sein. Jedoch ist es hier nur schwer möglich, dem Programm eine saubere Struktur zu geben. So entsteht sehr leicht der sogenannte Spaghetti-Code, wo beliebig von einem Befehl über das halbe Programm zum anderen Befehl gesprungen wird. Maschinensprache hat den Vorteil, daß man hier in keiner Weise eingeschränkt ist und der Programmcode bei richtiger Programmierung optimal schnell ist. Allerdings fällt vielen die Programmierung in Maschinensprache sehr schwer, da man sehr abstrakt denken muß.

Modula2 schließlich gehört zu den strukturierten Programmiersprachen. Der Ablauf der Programms ist schon an seiner äußeren Form erkennbar. Dies macht den Programmcode schnell durchschaubar, was sicherlich ein großer Vorteil ist. Der Nachteil ist, daß oft mehrere Befehle nötig sind, während in C beispielsweise ein einzelner Befehl reichen würde. Es ist also etwas mehr Schreibarbeit nötig, was sich unserer Meinung nach aber lohnt. Dies muß jedoch jeder für sich wissen. Lernen Sie Modula2 einfach in diesem Kurs kennen, und entscheiden Sie dann, ob Ihnen die Sprache zusagt oder nicht. Der Einstieg wird Ihnen in keiner anderen Sprache sonst so preiswert ermöglicht. Sie müssen lediglich für zirka 5 DM den PD-Compiler erwerben oder ihn einfach bei einem Freund kopieren und sich die AMIGA DOS kaufen – was Sie hoffentlich sowieso immer tun!

Die Installation

Zuerst einmal muß der Modula2-Compiler installiert werden. Da allerdings beim PD-Compiler keine Erklärung mitgeliefert wird, werden wir auch nur die Installation dieses Compilers erläutern. Wer im Besitz der kommerziellen Version ist, sollte im Handbuch nachschlagen. Alles, was Sie zur Installation brauchen, ist eine leere, formatierte Diskette und die Workbench. Booten Sie Ihren Rechner bitte mit der Workbenchdiskette, und legen Sie die Modula2-Diskette in das zweite Laufwerk. Besitzen Sie nur ein Laufwerk, so legen Sie die Modula2-Diskette in dieses Laufwerk. Lassen Sie sich nun das Inhaltsverzeichnis anzeigen, indem Sie auf dem Icon der Modula2-Diskette doppelklicken. Starten Sie das Programm 'DoMe' ebenfalls durch Doppelklick. Es erscheint ein Fenster, in dem gefragt wird, ob Sie mit einem oder zwei Laufwerken arbeiten. Klicken Sie hier entsprechend das jeweilige Gadget an. Nun werden Sie aufgefordert, die formatierte Diskette ins Startlaufwerk zu legen, was in der Regel das interne Laufwerk ist. Besitzen Sie noch ein zweites, so legen Sie die Workbenchdiskette ins zweite Laufwerk. Klicken Sie nun das Gadget Weitermachen an, woraufhin der Installationsvorgang beginnt. Falls Sie aufgefordert werden, eine Diskette zu wechseln, was bei Systemen mit nur einem Laufwerk garantiert vorkommt, so befolgen Sie die Anweisungen. Der Installationsvorgang dauert ungefähr 10 Minuten. Anschließend klicken Sie bitte eines der beiden Ende-Gadgets an. Starten Sie nun den CLI (er befindet sich auf der Workbenchdiskette), legen die Modula2-Diskette in das interne Laufwerk, und geben Sie ein:

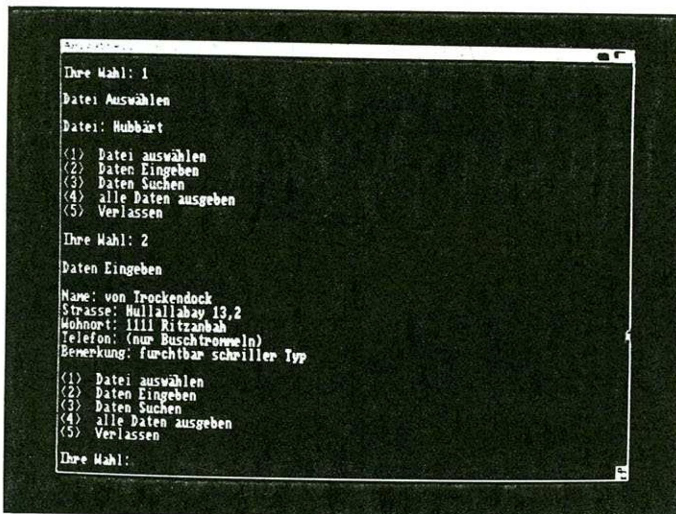
install DF0:

Dieser Befehl macht die Modula2-Diskette bootfähig. Der Installationsvorgang ist hiermit beendet. Sie können die Disketten aus den Laufwerken nehmen und einen Reset ausführen. Booten Sie nun von der erstellten Modula2-Diskette. Wir werden übrigens von nun an nur noch mit dieser Diskette arbeiten, sie ist unsere Ausgangsbasis für die nun folgenden Beispiele.

Arbeiten mit der Workbench

Der Modula2-Compiler bietet im Gegensatz zu den gängigen C-Compilern dem Anwender die Möglichkeit, alle Programme von der Workbench aus zu erstellen. Dies macht ihn gerade für Einsteiger sehr interessant. Natürlich können Sie ihn auch vom CLI aus benutzen. Wir werden Ihnen zuerst zeigen, wie Sie ein einfaches Programm von der Workbench aus erstellen. Klicken Sie bitte das Icon M2emacs an. Hierbei handelt es sich um den Editor, mit dem das Programm erstellt werden kann. Er ist wesentlich besser als der Editor ED des CLI. Außerdem kann er Fehler direkt anzeigen. Geben Sie nun bitte Listing 1 ein. Achten Sie dabei auf genaue Groß- und Kleinschreibung, denn Modula2 unterscheidet ganz genau zwischen großen und kleinen Buchstaben. Haben Sie das Listing eingegeben, so speichern Sie es als ASCII-Text ab. Wählen Sie dazu aus dem Menü 'Projekt' den Menüpunkt 'Sichern als' aus und geben Sie in der darauf erscheinenden Eingabezeile

m2amiga:HelloWorld.mod



Das Ergebnis der Bemühungen: eine kleine Dateiverwaltung in Modula2

ein und drücken Sie dann <Return>. Die Datei wird nun unter dem Namen 'Hello-World.mod' auf der Modula2-Diskette abgespeichert. Verlassen Sie nun den Editor durch Anklicken der Option 'Ende'. Laden Sie nun den Compiler 'm2c' durch Doppelklick, und geben Sie in dem erscheinenden Fenster

m2amiga:HelloWorld.mod

ein. Wenn Sie nun <Return> drücken, passiert erst einmal nichts. Das Laufwerk fängt zwar an zu laufen, doch erst nach einiger Zeit werden ein paar Zeilen ausgegeben. Doch dann erscheint eine Fehlermeldung. Wir müssen also zurück in den Editor und den Fehler beheben. Verlassen Sie zuerst den Compiler durch Drücken von <Return>. Das Fenster sollte daraufhin verschwinden. Laden Sie den Editor wie oben beschrieben, und wählen Sie den Menüpunkt 'Projekt/Öffne Datei' aus. Geben Sie als Namen

m2amiga:HelloWorld.mod

ein. Wählen Sie dann den Menüpunkt 'Lies Fehlerliste' aus dem Modula2-Menü des Editors. Eine Datei, die eine Auflistung der Fehler enthält, wird eingeladen. Lassen Sie sich nun den ersten Fehler durch 'Nächster Fehler' anzeigen. Der Cursor wird auf Zeile 8 positioniert, und in der Statuszeile erscheint der Text 'Bezeichner nicht vereinbart oder nicht sichtbar'. Ändern Sie Zeile 8 nun um in

WriteLn;

Was es mit der Fehlermeldung auf sich hat, wird noch an anderer Stelle beschrieben, wichtig ist erst mal, daß Sie den Umgang mit dem Modula2-System lernen. Speichern Sie nun diese Datei durch Auswahl des Menüpunktes 'Sichern' aus dem Projekt-Menü ab, verlassen Sie den Editor (Projekt/Ende), und starten Sie wieder den Compiler 'm2c' durch Anklicken seines Icons. Geben Sie hier nun erneut

m2amiga:HelloWorld.mod

ein. Das Laufwerk beginnt wieder zu laufen und nach einiger Zeit sollte das Prompt wieder erscheinen. Der Compiliervorgang ist hiermit abgeschlossen. Verlassen Sie den Compiler durch Drücken von <Return>. Sollte wieder eine Fehlermeldung erscheinen, gehen Sie wieder zurück in den Editor, und lassen Sie sich den Fehler anzeigen. Gehen Sie dabei analog zu dem Vorgehen vorhin vor. Haben Sie das Programm so eingegeben, wie es in diesem Heft abgedruckt ist, sollte der Compiliervorgang reibungslos klappen.
Der Compiler hat nun eine Objektdatei erstellt. Diese enthält zwar das übersetzte Programm, ist aber noch nicht ausführbar. Dazu muß sie erst mit einigen Modulen 'gelinkt' werden. Dieser Vorgang ist nötig, um das Programm zum Selbststart vorzubereiten. Linken ist ein gängiger Ausdruck in der Computerszene, der, wie so oft aus dem englichen stammt und sinngemäß 'verbinden' oder 'zusammenbinden' bedeutet. 'Linken' ist eine Arbeit, die der 'Linker' verrichtet.

Linken (hat nichts mit Übervorteilen zu tun)

Was geschieht jedoch beim Linken eines Programms? Um Amiga-Programme, gleich welcher Art, startfähig, also direkt ausführbar zu bekommen, ist eine Vorbereitung nötig. Da die Strukturen zum Selbststart überall gleich sind (nur sprachunterschiedlich) kann man diese Strukturen isolieren und dem eigenen Programm durch eben dieses Linken hinzufügen. Das bedeutet also, daß die Programmteile, die das Programm zum Selbststart bringen, für jedes andere ebenfalls einsetzbar sind. Dies vereinfacht die Sache ungemein, denn sonst müßten wir jedesmal die Startstrukturen neu schreiben und mitcompilieren. Statt dessen besitzen wir einen Linker, der dies für uns erledigt.
Starten Sie also den Linker durch Anklicken des Icons 'm2l'. Geben Sie in dem Fenster des Linkers nun die Zeile

m2amiga:HelloWorld.obj

ein. Das Laufwerk beginnt wieder zu laufen und nach einiger Zeit erscheint wieder das Prompt. Drücken Sie auch hier <Return>, um den Linker zu verlassen. Wir wollen das Programm nun einmal starten. Schließen Sie dazu das Fenster des Inhaltsverzeichnisses, und öffnen Sie es dann erneut. Sie werden nun einige neue Icons entdecken, darunter eines mit dem Namen HelloWorld. Eventuell müssen Sie den Inhalt des Fensters verschieben, damit Sie es entdecken. Starten Sie nun das Programm durch einen Doppelklick. Es wird ein Fenster geöffnet, in dem der Satz 'Hello World' ausgegeben wird. Nichts Aufregendes, aber dafür Ihr erstes selbstgeschriebenes Programm! Klicken Sie das Close-Gadget an, um das Fenster wieder zu schließen.

Arbeiten mit dem CLI

Für diejenigen, die etwas mehr Erfahrung mit dem CLI haben, möchten wir hier noch zeigen, wie man den Compiler vom CLI aus aufrufen kann. Dies funktioniert an sich analog zur Workbench, allerdings hat der Anwender hier noch die Möglichkeit, sich an das sogenannte Projektprinzip zu halten. Um mit dem CLI zu arbeiten, müssen Sie erst das CLI auf Ihre Modula2-Diskette kopieren. Schauen Sie in Ihrem DOS-Handbuch nach, wie man das macht. Wie Sie eben schon gemerkt haben, erzeugen der Compiler und der Linker einige Dateien. Dabei kennzeichnet immer die Endung einer Datei, um welche Daten es sich hierbei handelt. Dabei gelten für die Dateien folgende Endungen:

Endung	Datei	Verzeichnis
.mod	Sourcecode	txt
.modE	Fehlerdatei	txt
.obj	Objektdatei	obj
.ref	Referenzdatei	ref
.sym	Symboldatei	sym
keine	Programm	bin

Wie bereits erwähnt, bietet Modula2 dem Benutzer die Möglichkeit, für ein zu erstellendes Programm ein Projekt anzulegen. Dazu müssen verschiedene Verzeichnisse erzeugt werden. Zuerst müssen Sie ein CLI-Fenster öffnen und mittels

cd m2amiga:

das Verzeichnis wechseln. Wir müssen nun zunächst das Hauptverzeichnis für das zu entwickelnde Programm erstellen. Wir nennen es in diesem Falle HW (=World). Geben Sie also zur Erstellung folgenden Befehl ein:

makedir HW

Wechseln Sie dann in das Verzeichnis mittels

cd HW

und erstellen Sie nun die Verzeichnisse, in denen die verschiedenen Dateien abgelegt werden. Geben Sie dazu

makedir txt
makedir obj
makedir ref
makedir sym
makedir bin

ein. In diesen Verzeichnissen legt nun der Compiler automatisch die entsprechenden Dateien ab. Welche Datei in welchem Verzeichnis abgespeichert wird, können Sie obiger Tabelle entnehmen. Wir wollen nun noch einmal unser Hello-World-Programm erstellen. Kopieren Sie den Source-Code (so nennt man das Programm, welches wir im Editor geschrieben haben, also Listing 1) in das Verzeichnis 'txt' mit dem Befehl

copy m2amiga:World.
mod
txt

Starten Sie nun den Compiler durch Eingabe von

m2c

und geben Sie, wenn das Prompt erscheint, nur

txt/World.mod

an. Sie können sowohl das Verzeichnis als auch die Endung weglassen. Der Compiler sucht automatisch in dem Verzeichnis txt nach dieser Datei, compiliert sie dann und legt die Objektdatei in dem Verzeichnis obj ab und die Referenzdatei in dem Verzeichnis ref. Wäre ein Fehler aufgetaucht, so würde die Fehlerdatei unter dem Namen World.modE ebenfalls in dem Verzeichnis txt abgespeichert. Verlassen Sie nun den Compiler durch Drücken von <Return>, und starten Sie anschließend den Linker durch Eingabe von m2l. Geben Sie auch hier den Namen

obj/World.obj

an. Der Linker bearbeitet sie und legt das fertige Programm, welches keine Endung besitzt, in dem aktuellen Verzeichnis ab. Verlassen Sie den Linker durch Drücken von <Return>. Sie können das Programm nun durch Eingabe von 'World' starten. Diesmal wird kein Fenster geöffnet, sondern der Text direkt im CLI ausgegeben.
An dieser Stelle muß ich auch einmal auf die kommerzielle Version des Compilers eingehen. Hier gibt es nämlich einen Unterschied. Bei Version 3.3 muß nicht mehr das Verzeichnis 'txt' bzw. 'obj' angegeben werden. Auch die Endung kann wegfallen. So würde in unserem Beispiel die Eingabe von

World

sowohl beim Compiler als auch beim Linker reichen. Das entstandene Programm wird im Verzeichnis 'bin' und nicht im aktuellen Verzeichnis abgespeichert. Es wird also durch Eingabe von

bin/HelloWorld

gestartet.

Direktes Compilieren

Es ist ein bißchen umständlich, immer erst den Compiler bzw. Linker aufzurufen, diesen zu laden und dann erst die zu compilierende, bzw. linkende Datei zu bestimmen. Geben Sie einfach beim Aufruf direkt den Namen mit an, also beispielsweise

m2c txt/HelloWorld.mod

oder

m2l obj/HelloWorld.obj

ein. Die Datei wird dann direkt compiliert oder gelinkt; nach dem Vorgang wird der Compiler (oder der Linker) wieder verlassen. So muß nicht immer erst <Return> gedrückt werden, um eines der beiden Programme zu beenden. Bei der kommerziellen Version 3.3 müssen Sie angeben:

m2c HelloWorld.mod

bzw.

m2l HelloWorld.obj

Ansonsten sind die Compiler aber gleich, und ich werde im folgenden nur noch auf den PD-Compiler eingehen.

Eine kleine Dateiverwaltung – erste Schritte in der neuen Umgebung

An dieser Stelle möchte ich noch einmal ausführlicher auf die einzelnen Verzeichnisse und deren Bedeutung eingehen. Das Verzeichnis 'txt' enthält den Sourcecode und die Fehlerdatei. In dem Verzeichnis 'obj' wird die Objektdatei abgespeichert, die das compilierte, aber noch nicht ausführbare Programm enthält. Dieses wird in dem Verzeichnis 'bin' abgespeichert. Eine Datei des Verzeichnisses 'bin' ist immer direkt ausführbar. Die beiden Verzeichnisse 'ref' und 'sym' enthalten Dateien, mit denen wir erst einmal wenig zu tun haben. Im Verzeichnis 'sym' sind die Symboldateien abgespeichert. Dies sind Dateien, die noch

```
MODULE HelloWorld;

FROM InOut IMPORT WriteString, WriteLn;

BEGIN
  WriteString("Das erste Programm...");
  WriteLn;
END HelloWorld.
```

Abb. 1. Mit unserem ersten Programm in Modula2 lernen Sie mit dem Compiler umzugehen.

im Zusammenhang mit eigenen Modulen erzeugt werden. Hier sind sie erst einmal uninteressant. Im Verzeichnis 'ref' findet man Dateien, die Informationen für den Debugger enthalten. Ein Debugger ist ein Programm, das dem Anwender erleichtert, aus erstellten Programmen Fehler herauszufinden und diese zu beseitigen. Auch die Dateien in diesem Verzeichnis brauchen wir erst einmal nicht.

In den meisten Kursen ist es bisher so, daß die ersten erstellten Programme recht sinnlos sind und nur Demonstrationszwecken dienen. Dies läßt sich auch in diesem Kurs nicht vermeiden. Damit Sie aber sehen, daß mit Modula2 recht nützliche Programme erstellt werden können, haben wir eine kleine Dateiverwaltung hinzugefügt, deren Listing Sie im folgenden im Editor 'm2emacs' bitte abtippen. Wir werden Ihnen beim Erstellen der Projektdateien und dem Compilieren hier noch einmal Schritt für Schritt helfen. Die folgenden Beispiele müssen Sie aber dann selber erstellen.

Bei den längeren Beispielen, wie eben der Dateiverwaltung, ist die kommerzielle Version des M2Amiga-Programms allerdings Voraussetzung. Während die kleineren Beispiele mit der PD-Version compiliert werden können, ist die Bearbeitung längerer Dateien leider nicht möglich. Um Modula2 kennenzulernen, reicht die Mini-Version aus, um vernünftig programmieren zu können, leider nicht. Da Sie aber das 'vernünftige' Programmieren lernen sollen, müssen wir hier Kompromisse machen.

Den Source-Code und das fertige Programm finden Sie natürlich auch auf der Databox. Die Dateiverwaltung hat den Namen DVW. Wir wollen vom CLI aus nach dem Projektprinzip arbeiten. Sie müssen gegebenenfalls den CLI starten, danach zur Sicherheit erst einmal

cd m2amiga:

eingeben. Nun erstellen wir das Hauptverzeichnis namens DVW mit

makedir DVW

Wechseln Sie in dieses Verzeichnis mittels

cd DVW

und erstellen Sie nun die Unterverzeichnisse 'txt', 'obj', 'bin', 'ref' und 'sym'. Geben Sie dazu die Befehle zum Erstellen der Unterverzeichnisse (siehe oben) ein. Starten Sie nun den Editor durch

run m2emacs

und geben Sie Listing 2 ein. Speichern Sie es mittels 'Sichern als' unter dem Namen 'txt/DVW.mod' ab, und verlassen Sie wieder den Editor. Starten Sie nun den Compiler durch Eingabe von

m2c DVW.mod

Der Compiler wird geladen, und die Datei 'DVW.mod' wird compiliert. Sollten Fehler angezeigt werden, so starten Sie den Editor erneut, laden mit 'Projekt/Öffne Datei' und Eingabe von 'txt/DVW.-mod' die Datei ein, laden dann die Fehlerliste mittels 'Modula-2/Lies Fehlerliste' und wählen dann den Menüpunkt 'Nächster Fehler' aus. Vergleichen Sie die angegebene Zeile mit der Zeile im Listing, und verbessern Sie sie. Sollten Sie den Fehler nicht finden, so schauen Sie einmal auf die vorherige oder die nächste Zeile - oft fehlt auch dort nur ein Semikolon oder ähnliches. Wiederholen Sie das Anzeigen des nächsten Fehlers so lange, bis die "Fehlermeldung" 'kein (weiterer) Fehler gefunden' erscheint, und speichern Sie die Datei erneut ab (Projekt/Sichern). Kehren Sie zurück zum CLI, und starten Sie den Compiler erneut.

Wurde die Datei zufriedenstellend compiliert, muß Sie gelinkt werden. Geben Sie dazu

m2l DVW.obj

ein. Der Linker erstellt nun ein ausführbares Programm und speichert es in dem Verzeichnis 'bin' ab, wo Sie es starten können.

Die Modula2-Diskette

Als letztes möchten wir in diesem Kurs einmal durchgehen, was sich auf der Modula2-Diskette eigentlich befindet. Wir sind zwar schon in der Lage, die Installation durchzuführen und ein Programm zu compilieren. Auch wissen wir, welche Dateien auf dem Weg vom Source-Code bis zum fertigen Programm erzeugt werden und welchen Zweck sie erfüllen. Somit haben wir schon ein solides Grundwissen. Wir möchten es noch dadurch abrunden, daß wir Ihnen einige wichtige Dinge erläutern. So können Sie sich dann auch den Compiler auf die Festplatte kopieren oder selbständig schon einen Fehler beheben.

Lassen Sie sich einmal das Inhaltsverzeichnis der Modula2-Diskette anzeigen. Geben Sie dazu

dir m2amiga:

ein. Folgendes Inhaltsverzeichnis wird aufgelistet:

c
devs
l
libs
Modules
s
system
HW
DVW

Eigentlich sind alle Verzeichnisse bekannt. Sie kennen sie sicherlich schon von der Workbenchdiskette her. Das einzige neue Verzeichnis, welches hinzugekommen ist, heißt 'Modules'. Wir wollen dieses Verzeichnis einmal näher betrachten. Geben Sie dazu zuerst

cd m2amiga:Modules

und dann

dir

ein. Das Inhaltsverzeichnis wird aufgelistet. Es enthält die drei Unterverzeichnisse Interfaces, Libraries und System. Damit Sie den Sinn dieser Dateien verstehen, muß ich etwas ausholen. Wie der Name Modula2 schon sagt, hat die Sprache etwas mit verschiedenen Modulen zu tun. Ein Modul enthält verschiedene Programmchen, die eine be-

stimmte Aufgabe erledigen. So muß sich der Programmierer nicht mehr um die Programmierung dieser einzelnen Dinge kümmern, sondern muß nur über eine festgelgte Art ein Modul aufrufen. Dieses Modul läuft dann völlig unabhängig vom eigenen Programm ab und gibt, nachdem es seine Arbeit erledigt hat, die Kontrolle wieder an das aufrufende Programm zurück. In dem Verzeichnis 'Modules' sind einige solcher Module enthalten. Auch hier gibt es zwei Arten von Modulen. Einmal Objektmodule, die sich in den Verzeichnisses 'obj' befinden und dementsprechend von Diskette nachgeladen werden müssen, und Module, oder genauer gesagt Programmroutinen, die sich fest im Speicher (ROM) des Amiga befinden. Zu beiden Arten von Modulen muß jedoch festgelegt werden, wie Daten und Informationen zwischen aufrufendem und aufgerufenem Modul ausgetauscht werden. Das geschieht in den sogenannten Symboldateien, die sich in den Verzeichnissen 'sym' befinden. Gehen wir einmal mit dem Befehl

cd Libraries

in das Verzeichnis Libraries. Lassen Sie sich hier das Inhaltsverzeichnis ausgeben (dir), so werden nur zwei Unterverzeichnisse angezeigt. Nun lassen Sie sich den Inhalt der beiden Verzeichnisse mit

dir opt a

anzeigen. Die enthaltenen Dateien bestimmen, wie Daten zwischen dem Hauptprogramm und den entsprechenden Modulen ausgetauscht werden. Die Module selber

Die Bedienung der kleinen Datenverwaltung

An dieser Stelle möchte ich Ihnen eine kleine Anleitung geben, mit der Sie zufriedenstellend mit DVW arbeiten können. Ich habe DVW mit Absicht sehr einfach gehalten, damit das Listing nicht so lang ist und Sie somit nicht so viel abzutippen haben. Das Programm fängt die wichtigsten Fehler ab und bietet die Grundfunktionen einer Dateiverwaltung. Natürlich gibt es weitaus bessere Programme, aber dies ist ein Kurs und kein Programmierwettbewerb. DVW arbeitet diskettenorientiert, das heißt Datensätze werden direkt auf der Diskette abgespeichert. Dies ist im Falle eines Systemabsturzes sicherer.

Starten Sie zuerst DVW im CLI durch Eingabe von *DVW*

Starten Sie das Programm auf keinen Fall mit dem 'run'-Befehl. Der dabei entstehende Effekt ist aber sehenswert! Nach dem Start von DVW wird das Hauptmenü ausgegeben. Es umfaßt fünf Menüpunkte:

<1> *Datei auswählen*
<2> *Daten eingeben*
<3> *Daten suchen*
<4> *alle Daten ausgeben*
<5> *DVW verlassen*

Bevor wir Daten eingeben, müssen wir zuerst eine Datei bestimmen, in welcher die Daten abgespeichert werden sollen. Drücken Sie dazu <1>. Alle Eingaben müssen durch <Return> beendet werden. Sie werden nun nach dem Namen der Datei gefragt. Geben Sie hier probehalber einmal *ram:Test* ein. Die Datei wird erstellt und das Hauptmenü wird wieder angezeigt. Nun müssen wir Daten eingeben, denn was ist schon eine Datei ohne Daten? Wählen Sie also Menüpunkt 2 aus. Sie werden zunächst nach dem Namen der Person gefragt. Natürlich nimmt man hier zuerst den Namen seiner/(s) Liebsten. Nun muß die Straße eingegeben werden, anschließend folgt der Wohnort, die Telefonnummer und eine Bemerkung. Möchten Sie keine Bemerkung eingeben, so drücken Sie einfach <Return>. Haben Sie alle Daten eingegeben, so erscheint wieder das Hauptmenü. Wie das Ganze aussieht, ist in Bild 1 dargestellt. Sie können nun weitere Daten eingeben, indem Sie noch einmal bei Menüpunkt 2 beginnen.

DVW speichert übrigens die Dateien als ASCII-Text ab. Sie können Sie mit einem Editor (beispielsweise dem 'm2emacs') verändern oder mit dem Type-Befehl ausgeben lassen. Soweit zur kleinen Dateiverwaltung, probieren Sie ruhig noch ein bißchen aus. Ich hoffe, Sie können das Programm für kleine Archivierungsarbeiten gut gebrauchen.

befinden sich in dem Verzeichnis 'obj'. Die Dateien mit der Endung '.obj' enthalten die Befehle. Auf die einzelnen Module werden wir noch in den nächsten Kursen genauer eingehen. Wir möchten hier einmal schematisch aufzeigen, was der Compiler macht, wenn ein Modul aufgerufen wird. Nehmen wir an, aus dem Modul mit dem Namen 'InOut' wird die Routine 'WriteString' aufgerufen. Der Compiler schaut zuerst in der Datei 'InOut.sym' nach, ob die Daten im Programm richtig übergeben wurden. Ist dies nicht der Fall, so erzeugt er einen Fehler, ansonsten wird ein Befehl eingefügt, der den Linker veranlaßt, die Routine zu dem Hauptprogramm hinzuzulinken.

Das zur prinzipiellen Vorgehensweise beim Compilieren. Wenn Sie hier nicht alles genau verstanden haben, ist das auch nicht so schlimm. Im Laufe der nächsten Kursteile wird dieses Thema immer wiederholt. Die ausführlichen Erklärungen folgen später.

Nun zu den Unterschieden der verschiedenen Verzeichnisse. Im Verzeichnis 'Libraries' befinden sich Module zur generellen Arbeit mit Modula2. In dem Verzeichnis 'System' ebenfalls, während sich im Verzeichnis 'Interfaces' nur das Unterverzeichnis 'sym' befindet. Dies liegt daran, daß sich die Routinen, die diese Module enthalten, fest im Speicher (ROM) befinden. Es sind also nur die Schnittstellen zwischen Hauptprogramm und den Routinen, nicht aber die Routinen nötig, da diese sich ja schon im ROM befinden.
Einen letzten Blick wollen wir nun ins Verzeichnis s: werfen. Lassen Sie sich das Inhaltsverzeichnis einmal mit dem 'dir'-Befehl anzeigen.
Hier finden Sie zwei neue Dateien. Die eine nennt sich "Modula-2 Fehlermeldungen" und enthält die Texte der Fehler, die Sie sich im Editor anzeigen lassen können. Wir haben diese Datei also schon unbewußt beim Anzeigen der Fehler beim Beispiel 'HelloWorld' gebraucht. Die andere Datei hat den Namen 'm2path' und ist auch recht wichtig. Hier wird festgelegt, in welchen Verzeichnissen Modula2 nach den Objektdateien sucht.

Das Verzeichnis s:

In unserem Beispiel müßten das die drei Verzeichnisse sein, die sich im Unterverzeichnis Modules befinden. Lassen Sie sich diese Datei durch

type s:m2path

anzeigen, so wird dies bestätigt. Sie können diese Datei mit einem beliebigen Editor erweitern und ändern. Modula2 sucht seine Objekte immer in den hier angegebenen Verzeichnissen. Sollten Sie mit der hier erstellten Diskette arbeiten, so ist ein Ändern der Datei nicht nötig. Installieren Sie Modula2 allerdings auf einer Festplatte, so müssen Sie die beiden Dateien in das Verzeichnis s: der Festplatte kopieren und die Datei m2path entsprechend abändern. Hier das Schema zur Installation vom M2Amiga auf einer Festplatte:

– Erstellen Sie auf der Festplatte das entsprechende Unterverzeichnis,

– kopieren Sie Compiler, Linker und Editor in dieses Verzeichnis.

– Erstellen Sie das Verzeichnis Modules, und kopieren Sie alle Module der M2Amiga-Diskette in dieses Verzeichnis.

– Kopieren Sie die Dateien m2path und "Modula-2 Fehlermeldungen" in das Verzeichnis s:, und ändern Sie die Datei 'm2path' entsprechend Ihren Pfaden für die Module ab. Beachten Sie, daß Sie, bevor Sie mit Modula2 arbeiten können, unbedingt den Stack auf mindestens 20000 Byte setzen. Tun Sie das nicht, so stürzt der Compiler höchstwahrscheinlich ab. Sie können dazu den CLI-Befehl 'stack' benutzen:

stack 20000

Hiermit wären wir am Ende des ersten Kursteiles angelangt. Sie haben den Compiler und seine Bestandteile kennengelernt. Außerdem wissen Sie, wie Sie Programme compilieren und können die Dateiverwaltung DVW benutzen. Im nächsten Kurs werden wir dann mit den ersten Befehlen beginnen.

(jb)

Die Aufgabe

Wie das so üblich ist, wenn man was lernt, gibt es auch Hausaufgaben. Nun haben wir im ersten Teil ja noch nicht allzuviel Neues kennengelernt, aber etwas sollten Sie doch tun:
Nehmen Sie sich das kleine Testprogramm noch einmal vor und editieren Sie es im 'm2emacs'-Editor. Entfernen Sie das Semikolon hinter dem 'WriteString'-Befehl, und compilieren Sie das Programm. Achten Sie darauf, welche Fehlermeldung Ihnen gezeigt wird, und versuchen Sie zu ergründen, welche Funktion das Semikolon hat.

Listings

```
MODULE DVW;

FROM InOut IMPORT WriteString, WriteLn, ReadInt, ReadString,
                  WriteInt, WriteHex, WriteCard;
FROM SYSTEM IMPORT ADR;
FROM Dos IMPORT FileLockPtr, Lock, UnLock, FileHandlePtr,
                Open, Close, Write, Seek, Read, IoErr, Delay,
                Output, Input;
FROM Strings IMPORT Length;
FROM Str IMPORT Compare;
FROM Arts IMPORT Error, BreakPoint;

TYPE NameString = ARRAY[0..40] OF CHAR;
     StrasseString = ARRAY[0..40] OF CHAR;
     WohnortString = ARRAY[0..30] OF CHAR;
     TelefonString = ARRAY[0..20] OF CHAR;
     BemerkungString = ARRAY[0..60] OF CHAR;
     DateiName = ARRAY[0..40] OF CHAR;

     Datensatz = RECORD
                   Name : NameString;
                   Strasse : StrasseString;
                   Wohnort : WohnortString;
                   Telefon : TelefonString;
                   Bemerkung : BemerkungString;
                 END;

VAR AktDatensatz : Datensatz;
    Wahl : INTEGER;
    DateiHandle : FileHandlePtr;
    CLIOutput, CLIInput : FileHandlePtr;
    EndPosition : INTEGER;

PROCEDURE WriteLnString(String : ARRAY OF CHAR);
BEGIN
  WriteString(String);
  WriteLn;
```

Listing: DVW

```
END WriteLnString;

PROCEDURE LoescheDatensatz;
BEGIN
  WITH AktDatensatz DO
    Name:="";
    Strasse:="";
    Wohnort:="";
    Telefon:="";
    Bemerkung:="";
  END;
END LoescheDatensatz;

PROCEDURE Init;
BEGIN
  CLIOutput:=Output();
  CLIInput:=Input();
  EndPosition:=0;
  LoescheDatensatz;
  DateiHandle:=NIL;
END Init;

PROCEDURE Menu;
BEGIN
  WriteLnString("<1>  Datei auswaehlen");
  WriteLnString("<2>  Daten Eingeben");
  WriteLnString("<3>  Daten Suchen");
  WriteLnString("<4>  alle Daten ausgeben");
  WriteLnString("<5>  Verlassen");
  WriteLn;
  WriteString("Ihre Wahl: ");
END Menu;

PROCEDURE LiesString(DateiHandle : FileHandlePtr;
                     VAR String : ARRAY OF CHAR);

VAR i : INTEGER;
    dummy : LONGINT;

BEGIN
  i:=0;
  REPEAT
    dummy:=Read(DateiHandle, ADR(String[i]), 1);
```

Listing: DVW

```
      INC(i);
    UNTIL (String[i-1]=CHR(0AH));
    String[i]:=CHR(0);
  END LiesString;

  PROCEDURE DateiAuswaehlen(VAR DateiHandle : FileHandle
  Ptr;
                              VAR EndPosition : INTEGER);

  VAR Datei : DateiName;
      DateiLock : FileLockPtr;

  BEGIN
    IF (DateiHandle # NIL) THEN
      Close(DateiHandle);
      DateiHandle:=NIL;
    END;
    WriteLnString("Datei Auswaehlen");
    WriteLn;
    WriteString("Datei: ");
    ReadString(Datei);
    DateiLock:=Lock(ADR(Datei), -2);
    IF DateiLock=NIL THEN
      DateiHandle:=Open(ADR(Datei), 1006);
    ELSE
      DateiHandle:=Open(ADR(Datei), 1005);
    END;
    IF (DateiHandle # NIL) THEN
      EndPosition:=Seek(DateiHandle, 0, 1);
      EndPosition:=Seek(DateiHandle, 0, 0);
    ELSE
      WriteLn;
      WriteLnString("Datei konnte nicht ge-ffnet werden"
  );
    END;
    UnLock(DateiLock);
    WriteLn;
  END DateiAuswaehlen;

  PROCEDURE DatenEingeben(VAR DateiHandle : FileHandlePt
  r;
                            VAR EndPosition : INTEGER);

  VAR dummy : LONGINT;

  BEGIN
    IF DateiHandle#NIL THEN
      WITH AktDatensatz DO
        LoescheDatensatz;
        WriteLnString("Daten Eingeben");
        WriteLn;
        WriteString("Name: ");
        dummy:=Read(CLIInput, ADR(Name), 40);
        Name[dummy]:=CHR(0);
        WriteString("Strasse: ");
        dummy:=Read(CLIInput, ADR(Strasse), 40);
        Strasse[dummy]:=CHR(0);
        WriteString("Wohnort: ");
        dummy:=Read(CLIInput, ADR(Wohnort), 30);
        Wohnort[dummy]:=CHR(0);
        WriteString("Telefon: ");
        dummy:=Read(CLIInput, ADR(Telefon), 20);
        Telefon[dummy]:=CHR(0);
        WriteString("Bemerkung: ");
        dummy:=Read(CLIInput, ADR(Bemerkung), 60);
        Bemerkung[dummy]:=CHR(0);
        dummy:=Write(DateiHandle, ADR(Name), Length(Name
  ));
        dummy:=Write(DateiHandle, ADR(Strasse),
                     Length(Strasse));
        dummy:=Write(DateiHandle, ADR(Wohnort),
                     Length(Wohnort));
        dummy:=Write(DateiHandle, ADR(Telefon),
                     Length(Telefon));
        dummy:=Write(DateiHandle, ADR(Bemerkung),
                     Length(Bemerkung));
        EndPosition:=Seek(DateiHandle, 0, 1);
        EndPosition:=Seek(DateiHandle, 0, 0);
      END;
    ELSE
      WriteLnString("Bitte erst Datei auswaehlen!");
    END;
    WriteLn;
  END DatenEingeben;

  PROCEDURE DatenSuchen(DateiHandle : FileHandlePtr;
                        EndPosition : INTEGER);

  VAR Position : INTEGER;
      SuchName : NameString;
      dummy : LONGINT;
      Zeichen : CHAR;
      Gefunden : BOOLEAN;

  BEGIN
    IF DateiHandle#NIL THEN
      Gefunden:=FALSE;
      Position:=0;
      dummy:=Seek(DateiHandle, 0, -1);
      Zeichen:="J";
      WriteString("Geben Sie den gesuchten Namen ein: ")
  ;
      dummy:=Read(CLIInput, ADR(SuchName), 40);
```



Listing: DVW

```
      SuchName[dummy]:=CHR(0H);
      WriteLn;
      WriteString("Suche ");
      WHILE (Position<EndPosition) AND (CAP(Zeichen)<>"N
  ") DO
        WITH AktDatensatz DO
          WriteString(".");
          LiesString(DateiHandle, Name);
          LiesString(DateiHandle, Strasse);
          LiesString(DateiHandle, Wohnort);
          LiesString(DateiHandle, Telefon);
          LiesString(DateiHandle, Bemerkung);
          IF ((Compare(SuchName, Name))=0) THEN
            Gefunden:=TRUE;
            WriteLnString("  Gefunden!");
            WriteLn;
            WriteString("Name: ");WriteString(Name);
            WriteString("Strasse: ");WriteString(Strasse
  );
            WriteString("Wohnort: ");WriteString(Wohnort
  );
            WriteString("Telefon: ");WriteString(Telefon
  );
            WriteString("Bemerkung: ");WriteString(Bemer
  kung);
            WriteLn;
            WriteString("Weitersuchen? [j/n] ");
            dummy:=Read(CLIInput, ADR(Zeichen), 1);
            WriteLn;
          END;
          Position:=Seek(DateiHandle, 0, 0);
        END;
      END;
      IF NOT(Gefunden) THEN
        WriteLn;
        WriteLn;
        WriteLnString("Konnte den Namen leider nicht fin
  den!");
      END;
      Position:=Seek(DateiHandle, 0, 1);
    ELSE
      WriteLnString("Bitte erst Datei eingeben!");
    END;
    WriteLn;
  END DatenSuchen;

  PROCEDURE DatenAusgeben(DateiHandle : FileHandlePtr;
                          EndPosition : INTEGER);

  VAR i, Position : LONGINT;
      Zeichen : CHAR;

  BEGIN
    IF DateiHandle#NIL THEN
      Position:=Seek(DateiHandle, 0, -1);
      WITH AktDatensatz DO
        REPEAT
          LiesString(DateiHandle, Name);
          WriteString("Name: ");WriteString(Name);
          LiesString(DateiHandle, Strasse);
          WriteString("Strasse: ");WriteString(Strasse);
          LiesString(DateiHandle, Wohnort);
          WriteString("Wohnort: ");WriteString(Wohnort);
          LiesString(DateiHandle, Telefon);
          WriteString("Telefon: ");WriteString(Telefon);
          LiesString(DateiHandle, Bemerkung);
          WriteString("Bemerkung: ");WriteString(Bemerku
  ng);
          WriteLn;
          Position:=Seek(DateiHandle, 0, 0);
        UNTIL (Position=EndPosition);
      END;
    ELSE
      WriteLnString("Bitte erst Datei auswaehlen!");
    END;
  END DatenAusgeben;

  PROCEDURE Verlassen;
  BEGIN
    IF (DateiHandle#NIL) THEN
      Close(DateiHandle);
    END;
  END Verlassen;

  BEGIN
    Init;
    REPEAT
      Menu;
      ReadInt(Wahl);
      WriteLn;
      CASE Wahl OF
        1: DateiAuswaehlen(DateiHandle, EndPosition);|
        2: DatenEingeben(DateiHandle, EndPosition);|
        3: DatenSuchen(DateiHandle, EndPosition)|
        4: DatenAusgeben(DateiHandle, EndPosition);|
        5: Verlassen|
      ELSE
        WriteLn;
        WriteLnString("Falscher Menuepunkt!");
        WriteLn;
      END;
    UNTIL (Wahl=5);
  END DVW.
```

Listing: DVW

Karsten Krause

BRICKS

Wir lassen uns nicht einmauern!

Kennen Sie das russische Spiel Tetris, das vor nicht allzu langer Zeit in den Spielhallen für Furore sorgte? Nun, ab heute brauchen Sie Ihr kostbares Geld nicht mehr an den Spielautomaten zu verschleudern, denn mit unserem GFA-BASIC-Listing können Sie eine Tetris-Variante auch bequem zu Hause absolvieren – vorausgesetzt, Sie besitzen einen Amiga und GFA-BASIC.

Bricks basiert auf dem Prinzip des Spiels Tetris. Ziel ist dabei, von dem oberen Spielfeldrand herunterfallende Spielsteine zu geschlossenen horizontalen Reihen aufzuschichten.

Durch geschicktes Drehen der Spielsteine und rechtzeitiges Einsetzen an den entsprechenden Stellen werden die Reihen aufgebaut. Anfangs mag dies noch relativ einfach sein, doch wird die Fallgeschwindigkeit der herabfallenden Steine, die übrigens nicht immer die gleiche Form besitzen, immer schneller. Ist eine Reihe vollständig, also ohne Leerstellen, so verschwindet sie vom Spielfeld, und man hat wieder Platz zum Bauen, beziehungsweise zum Ausbessern von Lücken in tiefer gelegenen Reihen. Erreicht eine lückenhafte Mauer allerdings den oberen Spielfeldrand, so ist das Spiel verloren.

Die Steuerung des Spiels ist einfach:

Bewegt wird ein Stein mit einem Joystick in Port 2, wobei der Feuerknopf zum Drehen des Steines um 90 Grad dient. Drückt man den Joystick nach oben, so wird ein Pause-Modus aktiviert, man hat also Zeit für einen kleinen Kaffee zwischendurch. In das Spielgeschehen gelangt man wieder durch Druck auf den Feuerknopf. Durch Ziehen des Joysticks nach unten wird ein Stein abgesetzt. Weiterhin kann man sich mit der Taste “n“ den nächstfolgenden Stein im Fenster “NEXT BRICK“ anzeigen lassen.

Sicher dürfte es für den einen oder anderen ganz interessant sein, wie das Programm, das trotz seiner Schnelligkeit keinerlei Maschinensprache-Routinen verwendet, aufgebaut ist. Auch das können wir bieten: Hier eine kleine Beschreibung der 20 verwendeten Procedures, aus denen das Programm aufgebaut ist.

starten:
Es werden alle benötigten Variablen dimensioniert, die Variablen definiert, die Farben gesetzt (Farben in DATA-Zeilen) und der Highscore nachgeladen. Danach werden die Steine und ihre Drehpositionen über eine mathematische Formel errechnet und für ihre spätere Spritedarstellung vorbereitet.

menue_1:
Die benötigten Grafikroutinen werden in dieser Procedure aufgerufen.

menue_2:
Auswahl der Menüpunkte (Starten, Anleitung, Highscore & Beenden) und Scrollen der Laufschrift am unteren Bildrand, deren Text in der Variablen text$ in der Procedure starten definiert ist.

Sprung in das Label level, anleitung, highscore_anzeigen, oder beenden.

beenden:
Es erfolgt eine Sicherheitsabfrage. Bei einer positiven Antwort erfolgt der Sprung in den Workbench-Screen, ansonsten zurück ins Bricks-Menü.

anleitung:
Hier wird die Kurzanleitung angezeigt (für diejenigen, die es nicht abwarten können und diesen Text nicht lesen).

level:
An dieser Stelle kann die Spielstärke (1 - Anfänger, 6 - Profi) eingestellt werden (Anmerkung: Stufe 6 ist fast unspielbar). Danach erfolgt der Sprung zu spiel_starten

spiel_starten:
Der Stein wird durch das Zufallsprinzip ermittelt und die Steinfarbe gesetzt, danach wird der nächste Stein und die Startposition berechnet. Während des Spiels wird die Taste “n“ abgefragt. Das dient dazu, den nächsten Stein in einem eigenen Fenster anzuzeigen. Die Spielsteine werden durch die Sprites 2 und 3 angezeigt, der Stein in der Anzeige durch die Sprites 4 und 5. Die Farbe des fallenden Steines wird während des Fallens zyklisch verändert, um einen neuen Effekt zu erzielen, der im Spiel Tetris nicht vorhanden ist.

Beim Herabfallen der Steine wird der Joystick in Port zwei ständig abgefragt, was für eine exakte Steuerung sorgt. Ist ein Stein an der unteren Begrenzung angekommen, wird er durch die Procedure test durch einen PBOX-Befehl gezeichnet. Die Punktvergabe erfolgt beim Ankommen des Steines. Weiterhin wird ständig überprüft, ob eine Reihe vollständig ist. Ist dies der Fall, sackt sie nach unten ab und verschwindet vom Spielfeld.

Nach Ende des Spiels wird überprüft, ob ein neuer Highscore erreicht wurde. Wenn ja, erfolgt der Sprung zur Procedure neuer_high-score. Danach werden die Sprites ausgeschaltet und zum Hauptmenü verzweigt.

naechster_stein:
Anzeige des nächsten Steines im Fenster (Next Brick) durch die Sprites 4 und 5.

test:
Test, ob der Stein am unteren Spielfeldrand angekommen ist.

Procedures für den Bildschirmaufbau

menue_grafik1:
Aufbau der Menühintergrundgrafik über spezielle Procedures.

menue_grafik2:
Aufbau des Menüauswahlfeldes im Anzeigeteil der Hintergrundgrafik.

spielfeld_grafik:
Aufbau des Spielfeldes.

Speziell entwickelte Grafik-Procedures

shadow (x1,y1,x2,y2,mode):
Spezielle Grafik mit 3D-Effekt.

box (x1,y1,x2,y2,mode):
Spezielle Box-Routine mit Farbabstufung.

gitter:
Gitterhintergrund für Menü und Spielfeld.

color_cycle (nr):
Farbwechsel für die Spielsteine.

loeschen:
Löscheffekt für das Anzeigefeld im Menü.

grafik_loeschen:
Löscheffekt für den Grafikbildschirm (Menü & Spielfeld).

scroll_text:
Laufschrift Routine für das Menü.

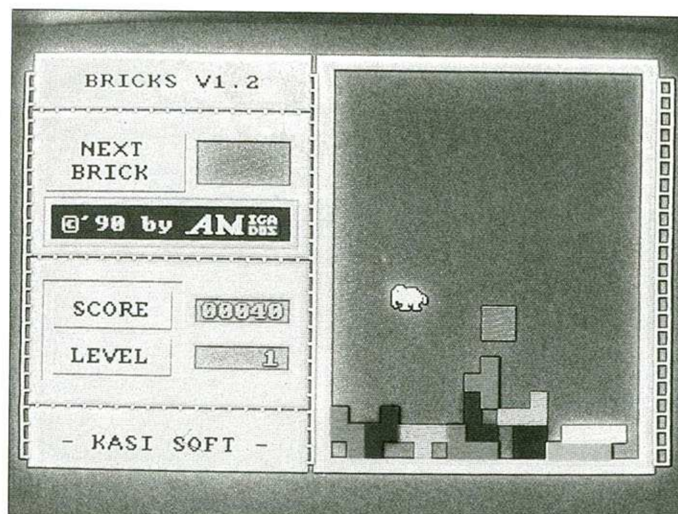


Bild 1. Bricks ist unsere Antwort auf das russische Geschicklichkeitsspiel Tetris

```
REM
REM Highscore für das Spiel
REM Bricks erstellen
REM
DIM p(10),a$(10)
REM
OPEN "o",#1,"-HighScore"
FOR i=1 TO 10
  PRINT #1,"........."
  PRINT #1,10
NEXT i
CLOSE #1
```

Abb. 1: MakeHighScore.lst

Diese Grafik-Procedures wurden extra für dieses Programm entwickelt. Sie sind natürlich vielfältig einsetzbar und können auch in eigene Programme eingebaut werden. Das Programm Bricks benötigt eine Datei namens "-Highscore" zum Abspeichern der Bestenliste. Ist diese Datei nicht im aktuellen Verzeichnis vorhanden, so meldet sich das Programm mit einer Fehlermeldung. Aus diesem Grund wurde das kleine Programm MakeHighScore.lst (siehe Abb. 1) entwickelt, das vor dem Aufruf von Bricks mit dem GFA-BASIC-Interpreter eingegeben und gestartet werden muß. (br)

Listings

```
REM ****************************************************
REM ***            B R I C K S  V1.2                 ***
REM ***    written by Karsten Krause (Kasi Soft)     ***
REM ***         (c) 1990 by DMV-Verlag               ***
REM ***           Sprache: GFA-BASIC                 ***
REM ****************************************************
REM
start
REM
PROCEDURE start
  zi=1
  st_anz=7
  max_pause=4
  xoff=144
  yoff=10
  xgr=20
  ygr=22
  OPENS 1,0,0,320,200,4,1
  OPENW #1,0,0,320,200,&H9,&H2
  DIM farbe(15,2),farbe2(15,2),stufen(15,2)
  DIM points(st_anz),deltana$(6),score$(10)
  DIM feld(ygr+2,xgr+2),steine(st_anz,3,3,3)
  DIM steine_gr(st_anz,1),steine$(st_anz,3,1)
  DIM a$(10),score(10)
  REM
  REM Text für die Laufschrift im Menü
  text$=" - KASI SOFT & AMIGA DOS präsentieren -"
  text$=text$+" *** B R I C K S ***"
  text$=text$+" geschrieben 1989/90 in GFA-BASIC"
  REM
  FOR a=0 TO 15
    READ farbe(a,0),farbe(a,1),farbe(a,2)
    SETCOLOR a,farbe(a,0),farbe(a,1),farbe(a,2)
  NEXT a
  DATA 0,0,0,14,0,0,15,14,0,0,8,0,0,13
  DATA 13,0,0,13,14,5,2,3,3,3,4,4,4
  DATA 5,5,5,7,7,7,9,9,9,10,10,10
  DATA 12,12,12,14,14,14,15,15,15
  REM
  COLOR 15,0
  TEXT 20,100,"Bitte die linke Maustaste drücken!"
  SETCOLOR 1,0,0,0
  REPEAT
  UNTIL MOUSEK
  TEXT 15,100,"Okay, noch etwas Geduld, ich rechne!"
  REM
  OPEN "i",#1,"-Highscore"
  FOR a=1· TO 10
    INPUT #1,a$(a),score(a)
  NEXT a
  CLOSE #1
  REM
  REM Steine berechnen
  FOR a=0 TO st_anz
    READ points(a),steine_gr(a,0),steine_gr(a,1)
    FOR b=0 TO steine_gr(a,0)
      READ a$
      FOR c=0 TO steine_gr(a,1)
        IF MID$(a$,c+1,1)<>" "
          steine(a,0,b,c)=-1
          steine(a,1,c,steine_gr(a,0)-b)=-1
          steine(a,2,steine_gr(a,0)-b,steine_gr(a,1)-c)=-1
          steine(a,3,steine_gr(a,1)-c,b)=-1
        ENDIF
      NEXT c
    NEXT b
  NEXT a
  DATA 4,1,2,"***"," * "
  DATA 4,1,2,"***","  *"
  DATA 4,1,2,"***","*  "
  DATA 4,1,2,"** "," **"
  DATA 4,1,2," **","** "
  DATA 6,1,2,"***","***
  DATA 4,1,1,"**","**"
  DATA 4,0,3,"****"
  FOR a=0 TO st_anz
    FOR b=0 TO 3
      FOR c=0 TO 3
        FOR d=0 TO 1
          FOR f=0 TO 7
            FOR e=0 TO 1
              IF steine(a,b,c,d*2+e)=-1
                steine$(a,b,d)=steine$(a,b,d)+CHR$(255)
              ELSE
                steine$(a,b,d)=steine$(a,b,d)+CHR$(0)
              ENDIF
            NEXT e
            steine$(a,b,d)=steine$(a,b,d)+CHR$(0)+CHR$(0)
          NEXT f
```

Listing: Bricks.lst

```
        NEXT d
      NEXT c
    NEXT b
  NEXT a
  REM
  menue_1
RETURN
REM
> PROCEDURE menue_1
  SETCOLOR 1,14,0,0
  menue_grafik1
  menue_grafik2
  menue_2
RETURN
> PROCEDURE menue_2
  PUT 30,30,menue$
  text=1
  a=0
  COLOR 0
  PBOX 0,189,320,200
  COLOR 10,0
  DO
    MOUSE mx,my,mk
    IF mk<>1 OR mx<62 OR mx>190
      GOTO weiter1
    ENDIF
    IF my>58 AND my<75
      level
    ELSE IF my>80 AND my<97
      anleitung
    ELSE IF my>102 AND my<119
      highscore_anzeigen
    ELSE IF my>124 AND my<141
      beenden
    ENDIF
    weiter1:
    scroll_text
  LOOP
RETURN
REM
> PROCEDURE spiel_starten
  score=0
  help=0
  ARRAYFILL feld(),0
  FOR a=0 TO ygr
    feld(a,0)=-1
    feld(a,xgr)=-1
  NEXT a
  FOR a=0 TO xgr
    feld(ygr,a)=-1
  NEXT a
  spielende=0
  next_st_nr=INT(RND(*st_anz+1))
  REPEAT
    ADD counter,0.1367    ! Fallgeschwindigkeit erhöhen
    col1=INT(RND*12)+2    ! Je höher der Wert,
    col2=INT(RND*12)+2    ! desto schneller...
    col3=INT(RND*12)+2
    col1r=1
    col2r=1
    col3r=1
    dreh=0
    st_nr=next_st_nr
    next_st_nr=INT(RND*(st_anz+1))
    x=INT(xgr/2-2)
    y=0
    xpos=x*8
    neuer_stein=0
    SETCOLOR 21,0,0,0
    movecounter=0
    mover1=0
    mover2=0
    moveanz=0
    pause=0
    score$=STRING$(5-LEN(STR$(score)),"0")+STR$(score)
    COLOR 10,0
    TEXT 90,127,score$
    TEXT 120,149,level$
    REPEAT
      VSYNC
      taste$=INKEY$             ! Nächsten Stein anzeigen
      IF taste$="n" AND help=0
        help=1
      ELSE IF taste$="n" AND help=1
        help=0
        SPRITE #4
        SPRITE #5
      ENDIF
      IF help=1
        naechster_stein
      ENDIF
      test(0)
      EXIT IF ok
      SPRITE #2,steine$(st_nr,dreh,0)
```

Listing: Bricks.lst

```
      SPRITE #3,steine$(st_nr,dreh,1)
      strigok=-1
      FOR a=0 TO 7
        color_cycle(21)
        SPRITE #2,xoff+xpos,yoff+y*8+a
        SPRITE #3,xoff+16+xpos,yoff+y*8+a
        IF movecounter>0
          ADD xpos,mover1
          DEC moveanz
          IF moveanz=0
            mover1=mover2
            DEC movecounter
            moveanz=4
          ENDIF
        ENDIF
        IF STRIG(1)
          IF strigok
            dreh2=dreh
            dreh=-(dreh+1)*(dreh<3)
            test(0)
            IF ok
              dreh=dreh2
            ENDIF
            strigok=0
          ENDIF
        ELSE
          strigok=-1
        ENDIF
        IF movecounter<2 AND pause=0
          SELECT STICK(1)
          CASE 1
            REPEAT
              color_cycle(21)
              FOR b=0 TO 250
              NEXT b
            UNTIL STICK(1)=0 AND STRIG(1)
            REPEAT
            UNTIL STICK(1)=0 AND STRIG(1)=0
          CASE 2
            test(0)
            IF NOT (ok)
              testende=0
              REPEAT
                INC y
                test(0)
                IF ok
                  DEC y
                  a=7
                  neuer_stein=-1
                  testende=-1
                  SPRITE #2,xoff+x*8+xcount,yoff+y*8+a
                  SPRITE #3,xoff+x*8+16+xcount,yoff+y*8
+a
                ENDIF
              UNTIL testende
            ENDIF
          CASE 4
            DEC x
            test(0)
            IF ok
              INC x
            ELSE
              INC movecounter
              IF movecounter=1
                moveanz=4
                mover1=-2
              ELSE
                mover2=-2
              ENDIF
            ENDIF
            pause=max_pause
          CASE 8
            INC x
            test(0)
            IF ok
              DEC x
            ELSE
              INC movecounter
              IF movecounter=1
                moveanz=4
                mover1=2
              ELSE
                mover2=2
              ENDIF
            ENDIF
            pause=max_pause
          ENDSELECT
        ENDIF
        IF pause>0
          DEC pause
        ENDIF
        FOR i=600-counter DOWNTO 0    ! Fallverzögerung
          REM
        NEXT i
      NEXT a
      INC y
    UNTIL neuer_stein
    EVERY -1 GOSUB game
    IF y=0
      spielende=-1
    ELSE
      ADD score,points(st_nr)
      col=INT(RND*7)+1
      SETCOLOR 21,farbe(col,0),farbe(col,1),farbe(col,2
)
      COLOR col,0
```

Listing: Bricks.lst

```
        test(-1)
      ENDIF
      IF y<ygr-4
        y1=y+4
      ELSE
        y1=ygr-1
      ENDIF
      FOR a=y1 DOWNTO y
        ok=-1
        FOR b=1 TO xgr-1
          IF NOT (feld(a,b))
            ok=0
          ENDIF
        NEXT b
        IF ok
          FOR b=a DOWNTO 1
            FOR c=1 TO xgr
              feld(b,c)=feld(b-1,c)
            NEXT c
          NEXT b
          FOR b=1 TO xgr-1
            feld(0,b)=0
          NEXT b
          SPRITE #2,STRING$(128,CHR$(0))
          SPRITE #3,STRING$(128,CHR$(0))
          ADD score,50
          FOR b=0 TO 7
            VSYNC
            SCROLL 0,1,xoff+9,yoff,xoff+xgr*8-1,yoff+a*8+
14
            VSYNC
          NEXT b
          INC a
        ENDIF
      NEXT a
    UNTIL spielende
    IF score>score(10)
      neuer_highscore
    ENDIF
    REPEAT
    UNTIL STRIG(1)
    SPRITE #2
    SPRITE #3
    SPRITE #4
    SPRITE #5
    grafik_loeschen
    menue_1
  RETURN
  > PROCEDURE naechster_stein
    SPRITE #4,steine$(next_st_nr,0,0)
    SPRITE #5,steine$(next_st_nr,0,1)
    SPRITE #4,94,48
    SPRITE #5,110,48
  RETURN
  > PROCEDURE level
    loeschen
    COLOR 11
    box(45,44,272,156,2)
    box(70,60,250,137,0)
    shadow(85,75,235,90,0)
    shadow(95,100,115,115,1)
    shadow(205,100,225,115,1)
    shadow(140,100,180,115,1)
    COLOR 5,12
    TEXT 101,110,"-"
    TEXT 211,110,"+"
    TEXT 144,110,"OKAY"
    COLOR 5,13
    TEXT 100,85,"Schwierigkeit:"
    DO
      MOUSE x,y,k
      IF y>114 OR y<101 OR k<>1
        GOTO weiter2
      ENDIF
      IF x>95 AND x<115 AND k=1
        DEC zi
      ELSE IF x>205 AND x<225 AND k=1
        INC zi
      ELSE IF x>140 AND x<180 AND k=1
        counter=zi*102
        spielfeld_grafik
      ENDIF
      weiter2:
      IF zi>6
        zi=6
      ELSE IF zi<1
        zi=1
      ENDIF
      level$=STR$(zi)
      TEXT 215,85,level$
      FOR a=1 TO 900
      NEXT a
    LOOP
  RETURN
  > PROCEDURE test(mode)
    LOCAL a,b,c,d
    ok=0
    a=steine_gr(st_nr,0)
    b=steine_gr(st_nr,1)
    IF dreh=1 OR dreh=3
      SWAP a,b
    ENDIF
    FOR c=y TO y+a
      FOR d=x TO x+b
        IF steine(st_nr,dreh,c-y,d-x)
          IF mode
```



Listing: Bricks.lst

```
          feld(c-1,d)=-1
          I d<xgr-1
            doff=0
          ELSE
            doff=1
          ENDIF
          PBOX xoff+d*8+1,yoff+c*8-1,xoff+d*8+8-doff,yo
ff+c*8+6
        ELSE
          IF feld(c,d)
            ok=-1
          ENDIF
        ENDIF
      ENDIF
    NEXT d
  NEXT c
  testende:
RETURN
REM
> PROCEDURE anleitung
  loeschen
  s=44
  COLOR 2
  TEXT s,48,"          ANLEITUNG             "
  TEXT s,68,"Dieses Spiel ist in Anlehnung"
  TEXT s,78,"an das Spiel Tetris geschrie-"
  TEXT s,88,"ben. Der Unterschied zum Ori-"
  TEXT s,98,"ginal liegt lediglich in der "
  TEXT s,108,"Spielfeldgröße und der Anzahl"
  TEXT s,118,"      der Spielsteine.        "
  COLOR 1
  TEXT s,128,"          Wichtig:            "
  COLOR 2
  TEXT s,138," Für Nebenwirkungen (Spiel-  "
  TEXT s,148,"sucht, Übermüdung etc.) wird "
  TEXT s,158," -KEINE- Haftung übernommen! "
  loeschen
  TEXT s,48,"  Ziel des Spieles ist es,   "
  TEXT s,58,"die herunterfallenden Steine "
  TEXT s,68," so zu ordnen, daß eine ge-  "
  TEXT s,78," schlossene Reihe entsteht.  "
  TEXT s,88,"Ist das geschafft, sackt die "
  TEXT s,98,"Reihe nach unten ab, und man "
  TEXT s,108,"  erhält 50 Zusatzpunkte.    "
  TEXT s,118," Sind jedoch Lücken in einer "
  TEXT s,128,"Reihe, muß man versuchen, die"
  TEXT s,138," darüberliegende(n) zum Ab-  "
  TEXT s,148,"sacken zu bringen. Dann ent- "
  TEXT s,158,"steht erneut die Möglichkeit,"
  loeschen
  TEXT s,52," die unvollständige Reihe zu "
  TEXT s,62,"    schließen. Erhebt sich   "
  TEXT s,72,"die Mauer bis zur oberen Be- "
  TEXT s,82,"grenzung, ist das Spiel ver- "
  TEXT s,92,"           loren.            "
  TEXT s,102,"Für besonders gute Leistungen"
  TEXT s,112,"gibt es dann noch einen HIGH-"
  TEXT s,122," SCORE, in dem die besten 10 "
  TEXT s,132," Spieler gespeichert werden. "
  TEXT s,142," Die Eingabe des Namens darf "
  TEXT s,152,"max. 10 Buchstaben beinhalten"
  loeschen
  TEXT s,48,"    Und so wird gespielt:    "
  TEXT s+4,68,"Man nehme einen Joystick in"
  TEXT s-1,78,"   Port 2 und los geht's    "
  COLOR 14
  TEXT s,93,"links/rechts - Stein bewegen "
  TEXT s,103,"nach unten   - Stein absetzen"
  TEXT s,113,"nach oben    - Pause         "
  TEXT s,123,"Feuerknopf   - Stein drehen  "
  TEXT s,133,"               & Pause aus   "
  TEXT s,143,"Taste < N >  - Steinanzeige  "
  COLOR 2
  TEXT s+4,158,"... Übung macht den Meister."
```



Listing: Bricks.lst

```
  loeschen
  menue_2
RETURN
REM
> PROCEDURE neuer_highscore
  COLOR 13,0
  PBOX 0,104,320,136
  COLOR 0,0
  PBOX 0,106,320,134
  COLOR 2,0
  TEXT 10,116,"Neuer Highscore! Herzlichen Glühwurm!"
  eingabe:
  PRINT AT(0,14);""
  INPUT "      Name : ",name$
  IF LEN(name$)>10 THEN
    TEXT 5,116,"Bitte nicht mehr als 10 Buchstaben!"
    PRINT AT(12,15);"                              "
    GOTO eingabe
  ENDIF
  FOR a=1 TO 10
    ok=0
    IF score>score(a)
      FOR b=9 DOWNTO a
        a$(b+1)=a$(b)
        score(b+1)=score(b)
      NEXT b
      score(a)=score
      a$(a)=name$
      ok=-1
      OPEN "o",#1,"-Highscore"          ! Neuen Highscore
      FOR i=1 TO 10                     ! speichern
        PRINT #1,a$(i)
        PRINT #1,score(i)
      NEXT i
      CLOSE #1
    ENDIF
    EXIT IF ok=-1
  NEXT a
RETURN
> PROCEDURE highscore_anzeigen
  loeschen
  COLOR 14,0
  TEXT 125,50,"HIGHSCORE"
  FOR i=1 TO 10
    score$(i)=STRING$(5-LEN(STR$(score(i))),"0")+STR$(s
core(i))
    COLOR i,0
    TEXT 58,54+i*10,a$(i)
    TEXT 215,54+i*10,score$(i)
  NEXT i
  loeschen
  menue_2
RETURN
REM
> PROCEDURE beenden
  loeschen
  ab$=""
  TEXT 50,89,"Sind Sie ganz sicher (j/n) ?"
  abfrage:
  ab$=INKEY$
  IF ab$<>"j" AND ab$<>"n"
    GOTO abfrage
  ELSE IF ab$="n"
    menue_2
  ENDIF
  grafik_loeschen
  CLOSEW #1
  CLOSES 1
  SYSTEM
RETURN
REM
REM ---------- Bildschirmaufbau ----------
REM
> PROCEDURE menue_grafikl
```

Listing: Bricks.lst

```
   CLS
   COLOR 0
   PBOX 0,0,320,200
   gitter
   box(0,10,319,189,2)
   box(36,35,281,165,2)
   box(104,5,216,22,0)
   COLOR 10
   COLOR 7,13
   TEXT 112,17,"BRICKS  V1.2"
RETURN
> PROCEDURE menue_grafik2
   COLOR 5,0
   PBOX 40,39,277,161
   COLOR 9
   PBOX 48,156,270,45
   COLOR 7
   BOX 52,49,266,152
   FOR i=57 TO 123 STEP 22
     shadow(62,i,190,i+17,1)
   NEXT i
   shadow(201,57,257,140,1)
   COLOR 5,12
   TEXT 75,70,"SPIEL STARTEN"
   TEXT 92,92,"ANLEITUNG"
   TEXT 92,114,"HIGHSCORE"
   TEXT 101,136,"BEENDEN"
   TEXT 214,75,"KASI"
   TEXT 214,85,"SOFT"
   TEXT 227,99,"&"
   TEXT 210,116,"AMIGA"
   TEXT 218,128,"DOS"
   COLOR 0,0
   GET 30,30,280,167,menue$
RETURN
> PROCEDURE spielfeld_grafik
   grafik_loeschen
   COLOR 0
   PBOX 0,0,320,200
   gitter
   COLOR 15
   LINE 319,11,319,195
   box(145,0,311,200,2)
   box(149,4,307,196,1)
   box(4,0,141,29,0)
   box(4,31,141,99,0)
   box(4,101,141,169,0)
   box(4,171,141,200,0)
   shadow(10,41,81,69,0)
   shadow(10,111,81,159,1)
   shadow(15,115,75,133,0)
   shadow(15,137,75,155,0)
   box(83,115,136,133,1)
   box(83,137,136,155,1)
   box(83,41,136,69,1)
   box(10,72,136,95,2)
   COLOR 5,13
   PBOX 14,76,132,91
   COLOR 5,13
   TEXT 25,127,"SCORE"
   TEXT 25,149,"LEVEL"
   TEXT 30,18,"BRICKS V1.2"
   TEXT 20,190,"- KASI SOFT -"
   TEXT 29,52,"NEXT"
   TEXT 24,63,"BRICK"
   COLOR 4,5
   TEXT 20,87,"⌐┤ 90 by"
   REM                                AMIGA DOS  LOGO
   COLOR 1,5
   PBOX 112,88,126,83
   COLOR 15,5
   LINE 82,88,87,88
   LINE 84,88,92,80
   LINE 85,88,93,80
   LINE 88,85,92,85
   LINE 91,88,94,88
   LINE 97,88,100,88
   LINE 106,88,109,88
   LINE 113,80,115,80
   LINE 113,83,115,83
   LINE 114,80,114,83
   LINE 118,80,120,80
   LINE 117,81,117,82
   LINE 118,83,119,83
   LINE 122,83,122,81
   LINE 123,80,124,80
   LINE 125,81,125,83
   LINE 122,82,125,82
   PBOX 98,88,99,80
   PBOX 92,80,93,88
   PBOX 107,88,108,80
   PLOT 120,82
   PLOT 97,80
   PLOT 109,80
   FOR a=82 TO 84
     LINE 100,a,104,a+4
     LINE 107,a,104,a+4
   NEXT a
   COLOR 15,1
   LINE 113,85,113,88
   LINE 113,85,115,85
   LINE 113,88,115,88
   LINE 116,86,116,87
   LINE 122,85,125,85
   LINE 122,88,125,88
   LINE 123,86,124,87
```



Listing: Bricks.lst

```
   BOX 118,85,120,88
   REM
   spiel_starten
RETURN
REM
REM ------------- Grafik Tools -----------
REM
> PROCEDURE shadow(x1,y1,x2,y2,mode)
   IF mode=1 THEN
     COLOR 7
     BOX x1,y1,x2-1,y2-1
     COLOR 14
     BOX x1+1,y1+1,x2,y2
     COLOR 12
     PBOX x1+1,y1+1,x2-1,y2-1
   ELSE
     COLOR 14
     BOX x1,y1,x2-1,y2-1
     COLOR 7
     BOX x1+1,y1+1,x2,y2
     COLOR 13
     PBOX x1+1,y1+1,x2-1,y2-1
   ENDIF
RETURN
> PROCEDURE box(x1,y1,x2,y2,mode)
   FOR a=0 TO 3
     IF mode=0 OR mode=2
       COLOR 9+a
     ELSE
       COLOR 13-a
     ENDIF
     BOX x1+a,y1+a,x2-a,y2-a
   NEXT a
   IF mode=0
     COLOR 13
   ELSE IF mode=1
     COLOR 0
   ENDIF
   IF mode<2
     PBOX x1+4,y1+4,x2-4,y2-4
   ENDIF
RETURN
> PROCEDURE gitter
   COLOR 15,0
   FOR a=0 TO 320 STEP 8
     LINE a,11,a,195
   NEXT a
   FOR a=11 TO 197 STEP 8
     LINE 1,a,319,a
   NEXT a
   COLOR 0,0
RETURN
> PROCEDURE color_cycle(nr)
   SETCOLOR nr,col1,col2,col3
   ADD col1,col1r
   ADD col2,col2r
   ADD col3,col3r
   IF col1=14 OR col1=2
     col1r=-col1r
   ENDIF
   IF col2=14 OR col2=2
     col2r=-col2r
   ENDIF
   IF col3=14 OR col3=2
     col3r=-col3r
   ENDIF
RETURN
> PROCEDURE loeschen
   REPEAT
   UNTIL MOUSEK
   COLOR 0
   PBOX 0,189,320,200
   FOR i=0 TO 117
     COLOR 15
     PBOX 42+i,39,42+i,162
     PBOX 276-i,39,276-i,162
     COLOR 0
     PBOX 41+i,39,41+i,162
     PBOX 277-i,39,277-i,162
   NEXT i
   PBOX 158,39,162,162
   COLOR 2
RETURN
> PROCEDURE grafik_loeschen
   COLOR 0
   FOR a=0 TO 320
     LINE a,0,320-a,200
     LINE 0,200-a*0.625,320,a*0.625
   NEXT a
RETURN
> PROCEDURE scroll_text
   IF a=0
     TEXT 311,198,MID$(text$,text,1)
     INC text
     IF text>LEN(text$)
       text=1
     ENDIF
   ENDIF
   VSYNC
   SCROLL -1,0,0,192,319,199
   INC a
   IF a=8
     a=0
   ENDIF
RETURN
```

Listing: Bricks.lst

Garry Glendown

SprEd – das Sprite-Editor-Projekt

Teil 6: Jetzt kommt Farbe ins Spiel...

Unser Sprite-Editor funktioniert zwar schon recht gut, doch besteht momentan noch ein kleines Problem: Man ist auf die Farben beschränkt, die im Programm festgelegt sind. Da das natürlich nicht so bleiben darf, widmen wir uns im heutigen Teil einer Farbrequester-Routine.

Bei der Programmierung wurde streng darauf geachtet, daß der Requester möglichst flexibel gestaltet ist, so daß er auch in eigenen Programmen verwendet werden kann. Dies wurde dadurch ermöglicht, daß alle Variablen, die vom Requester global genutzt werden, dem Hauptprogramm nicht bekannt gemacht werden. Auf diese Weise braucht man nicht darauf zu achten, daß sich Variablennamen überschneiden.

Des weiteren verfügt der Farbrequester über die Fähigkeit, sowohl mit 16, wie es in SprEd benötigt wird, als auch mit zwei, vier oder acht Farben zu arbeiten. Eine Erweiterung auf 32 Farben ist zudem recht leicht möglich.

Doch wie ruft man den Requester nun auf? Kern des ganzen Programms ist die Routine 'ColReq'. Sie wird mit zwei Parametern aufgerufen, die zum einen den Zeiger auf die zu editierende Palette (USHORT *) und zum anderen die Anzahl der Bitplanes darstellen. Letzteres darf lediglich Werte zwischen eins und vier annehmen. Bevor der Requester geöffnet wird, wird zunächst überprüft, ob sowohl Intuition als auch die Graphics-Library geöffnet sind, beziehungsweise, ob die Anzahl der Bitplanes zulässig ist.

Um eine flexible Bearbeitung verschiedener Farb-Anzahlen zu ermöglichen, wurde für die Farbauswahl nur auf ein Gadget zurückgegriffen, so daß nur eine Umrechnung der Maus-Position in eine Farbe vorgenommen werden muß. Außerdem spart man Zeit beim Aufbau des Requesters.

Natürlich kann man auch nicht auf eine gewisse Effekthascherei verzichten. Der Requester, der ohnehin in einem eigenen Screen erscheint, wird beim Starten sanft in den Bildschirm gescrollt, wird die Arbeit mit der Farbpalette beendet, wird der Screen ebenso wieder entfernt.

Im linken Teil des Screens befinden sich vier Gadgets: Drei Schieberegler, mit denen man die einzelnen Komponenten Rot, Grün und Blau modifizieren kann, sowie ein String-Gadget, in dem die aktuelle Farbeinstellung in Hexadezimalform angezeigt wird. Ferner besteht die Möglichkeit, direkt einen Farbwert ebenfalls als 'Hex-Zahl' einzugeben. Fehleingaben wie '!' oder 'K' werden dabei insoweit ignoriert, daß die Zeichen, die kleiner als '0' sind, als 0 interpretiert werden, alle Zeichen, die größer als 'F' sind, als 15.

Die Bedienung – einfach und übersichtlich

Mittels 'Reset' kann die aktuelle Palette wieder in den Ausgangszustand gebracht werden. Das 'OK'-Gadget bestätigt die Änderung, und die Arbeit im Farbrequester wird beendet. Wählt man das Close-Gadget des Fensters an, so wird der Requester verlassen, ohne daß die Änderungen an das aufrufende Programm weitergegeben werden.

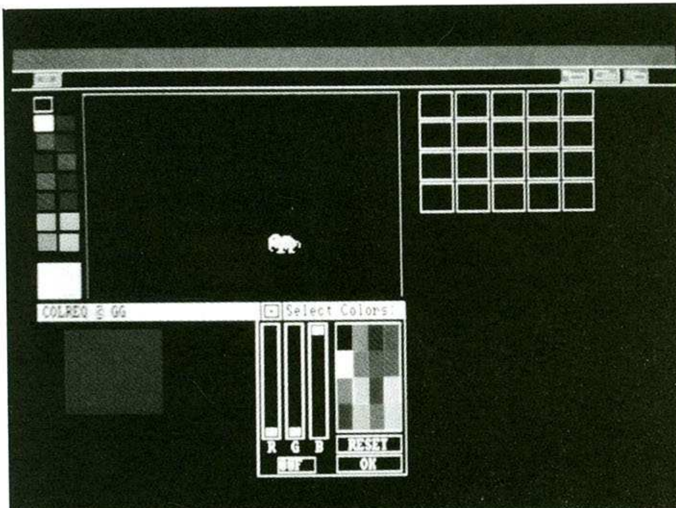
ColReq kann in einem Programm ganz normal aufgerufen werden, beim Linken muß lediglich das compilierte '.o'-File hinzugelinkt werden.

Nichts ist unmöglich – Besonderheiten

Außerdem ist zu beachten, daß ein Return-Wert von '-1' bedeutet, daß während des Aufrufs irgendein Fehler (zu wenig Speicher, fehlende Library, etc.) aufgetreten ist. Wurde der Requester mit dem Close-Gadget beendet, so wird ein Wert von 2 zurückgegeben, beim Beenden mit dem OK-Gadget hingegen der Wert '1'.

(br/vb)

```
# Alle Includes wurden
mit der Option ' + hlib:all.
pre' vorcompiliert
OBJ =  spred.o part2.o
       part3.o
       part4.o fertig.o
       title.o colreq.o
ALL: spred coltest
.c.o:
  cc $*.c + ilib:all.pre
spred: $(OBJ)
  ln + q + cd $(OBJ) -la -lm
-lc
coltest: colreq.o coltest.o
  ln + q coltest.o colreq.o
  -lc
```



Die Farbauswahl leichtgemacht mit dem Farbrequester

Änderungen im File spred.c (Ausgabe 2'90)

Zeile 29:

```
Alt: struct UCopList *myCop,*origCop;
Neu: struct UCopList *myCop,*tcl,*origCop;
```

Zeilen 499/500:

```
Alt:    view = ViewAdress();
        vp = view->ViewPort;
Neu:    /* view = ViewAdress();
        vp = view->ViewPort;*/
        Evp = &(scr->ViewPort;
```

Zeile 505 bis 530 löschen und ersetzen durch:

```
BuildCopList(myCop);
vp->UCopIns = myCop;
```

Nach Zeile 533 Listing 1 einfügen

Zeile 540-543 ersetzen durch:

```
struct CopList *cl,*cl1;
if      (myCop) {
        Evp->UCopIns = origCop;
        RethinkDisplay();
        FreeCopList(myCop->FirstCopList);
        FreeMem(myCop,(long)sizeof(struct UCopList));
}
```

Änderungen in Listing Part2.c (Ausgabe 3'90)

Nach Zeile 12 einfügen:

```
extern USHORT SecPal[];
```

Zeile 106 ersetzen durch:

```
{
int t;
    NewCopList(-1);
     t = ColReq(SecPal,4);
    NewCopList(0);
}
```

Listings



```
NewCopList(num)
int num;
{
  tcl=myCop;
  if (!(myCop=AllocMem((long)sizeof(struct UCopList),
    MEMF_PUBLIC|MEMF_CLEAR)))
      puts("Mem");
  else {
    BuildCopList(myCop,num);
    vp->UCopIns=myCop;
    RethinkDisplay();
    FreeCopList(tcl->FirstCopList);
    FreeMem(tcl,(long)sizeof(struct UCopList));
  }
}

BuildCopList(cl,num)
struct UCopList *cl;
int num;
{
int t;
  CWAIT(cl,0L,14L);
  CMOVE(cl,custom.color[0],(long)0xfff);
  CWAIT(cl,1L,14L);
  CMOVE(cl,custom.color[0],(long)0x000);

  for (t=1;t<16;t++)
    CMOVE(cl,custom.color[t],(long)(Palette[t]));

  CWAIT(cl,13L,14L);
  CMOVE(cl,custom.color[0],(long)0xFFF);
  CWAIT(cl,14L,14L);

  for (t=0;t<16;t++)
    CMOVE(cl,custom.color[t],(long)SecPal[t]);

  if (num!=-1) {
    CWAIT(cl,185L,14L);
    CMOVE(cl,custom.color[0],(long)0xfff);
    CWAIT(cl,186L,14L);
    CMOVE(cl,custom.color[0],0L);

    for (t=1;t<16;t++)
      CMOVE(cl,custom.color[t],(long)(Palette[t]));
  }

  CEND(cl);
}
```

Listing 1: Änderungen im File spred.c

```
/*******************************************************/
/*                      ColReq.c                       */
/*     (C) 1990 by G. Glendown/AMIGA DOS               */
/*     Sprache: Aztec C                                */
/*******************************************************/

static struct NewScreen scrdef = {
  0,296,379,105,
  1,  /* jeweils aendern */
  0,1,HIRES,CUSTOMSCREEN,NULL,
  (UBYTE *)"COLREQ (c) GG & DMV",NULL,NULL};


static USHORT Palette[] = {
  0x444,0x666,0x777,0x999,
  0xAAA,0xCCC,0xDDD,0xFFF,
  0x444,0x666,0x777,0x999,
  0xAAA,0xCCC,0xDDD,0xFFF};

static UBYTE UNDOBUFFER[4];

static SHORT BorderVectors1[] =
  {0,0,65,0,65,10,0,10,0,0};
static struct Border Border1 =
  {-1,-1,1,0,JAM1,5,BorderVectors1,NULL};

static struct TextAttr TOPAZ80 =
  {(STRPTR)"topaz.font",TOPAZ_EIGHTY,0,0};
static struct IntuiText IText1 =
  {1,0,JAM2,23,1,&TOPAZ80,(UBYTE *)"OK",NULL};

static struct Gadget Gadget7 =
  {NULL,81,93,64,9,GADGHBOX,RELVERIFY,
  BOOLGADGET,(APTR)&Border1,NULL,&IText1,
  NULL,NULL,21,NULL};

static SHORT BorderVectors2[] =
  {0,0,65,0,65,10,0,10,0,0};
static struct Border Border2 =
  {-1,-1,1,0,JAM1,5,BorderVectors2,NULL};

static struct IntuiText IText2 =
  {1,0,JAM2,11,1,&TOPAZ80,(UBYTE *)"RESET",NULL};

static struct Gadget Gadget6 =
  {&Gadget7,81,81,64,9,GADGHBOX,RELVERIFY,BOOLGADGET,
  (APTR)&Border2,NULL,&IText2,NULL,NULL,20,NULL};
```

Listing: Colreq.c

```

static SHORT BorderVectors3[] =
  {0,0,65,0,65,65,0,65,0,0};
static struct Border Border3 =
  {-1,-1,1,0,JAM1,5,BorderVectors3,NULL};

static struct Gadget Gadget5 =
  {&Gadget6,81,13,64,64,GADGHBOX|GADGHIMAGE,
  RELVERIFY,BOOLGADGET,(APTR)&Border3,
  NULL,NULL,NULL,NULL,10,NULL};

static UBYTE Gadget4SIBuff[4] ="000";
static struct StringInfo Gadget4SInfo =
  {Gadget4SIBuff,UNDOBUFFER,0,4,
  0,0,0,0,0,0,0,0,NULL};
#define COLTXT  Gadget4SIBuff
#define COLTXTG  Gadget4

static SHORT BorderVectors4[] =
  {0,0,37,0,37,9,0,9,0,0};
static struct Border Border4 =
  {-1,-1,1,0,JAM1,5,BorderVectors4,NULL};

static struct Gadget Gadget4 =
  {&Gadget5,22,94,36,8,NULL,RELVERIFY,
  STRGADGET,(APTR)&Border4,NULL,NULL,NULL,
  (APTR)&Gadget4SInfo,14,NULL};

static struct PropInfo Gadget3SInfo =
  {AUTOKNOB|FREEVERT,-1,-1,-1,4095};

static struct Image Image1 =
  {0,61,11,5,0,NULL,0x0000,0x0000,NULL};

static struct Gadget Gadget3 =
  {&Gadget4,53,12,19,70,NULL,RELVERIFY,
  PROPGADGET,(APTR)&Image1,NULL,NULL,NULL,
  (APTR)&Gadget3SInfo,13,NULL};
#define BGADG  Gadget3

static struct PropInfo Gadget2SInfo =
  {AUTOKNOB|FREEVERT,-1,-1,-1,4095};

static struct Image Image2 =
  {0,61,11,5,0,NULL,0x0000,0x0000,NULL};

static struct Gadget Gadget2 =
  {&Gadget3,29,12,19,70,NULL,RELVERIFY,
  PROPGADGET,(APTR)&Image2,NULL,NULL,NULL,
  (APTR)&Gadget2SInfo,12,NULL};
#define GGADG  Gadget2

static struct PropInfo Gadget1SInfo =
  {AUTOKNOB|FREEVERT,-1,-1,-1,4095};

static struct Image Image3 =
  {0,61,11,5,0,NULL,0x0000,0x0000,NULL};

static struct IntuiText IText3 =
  {1,0,JAM2,6,71,&TOPAZ80,(UBYTE *)"R  G  B",NULL};

static struct Gadget Gadget1 =
  {&Gadget2,5,12,19,70,NULL,RELVERIFY,
  PROPGADGET,(APTR)&Image3,NULL,&IText3,NULL,
  (APTR)&Gadget1SInfo,11,NULL};
#define RGADG  Gadget1

#define GadgetList1 Gadget1

static struct NewWindow windef = {227,0,152,105,0,1,
  GADGETUP|CLOSEWINDOW|VANILLAKEY,
  WINDOWCLOSE|ACTIVATE|RMBTRAP|NOCAREREFRESH,
  &Gadget1,NULL,(UBYTE *)"Select Colors:",NULL,
  NULL,5,5,-1,-1,CUSTOMSCREEN};

static struct Screen *scr;
extern struct Screen *OpenScreen();

static struct Window *win;
extern struct Window *OpenWindow();

extern struct IntuitionBase *IntuitionBase;
extern struct GfxBase *GfxBase;

static struct RastPort *rp;
static struct ViewPort *vp;

static int sx[]={0,64,32,32,16};
static int sy[]={0,32,32,16,16};
static int fx[]={0,0,2,4,4};

/* Aktuelle Farbnummer, Farbwert */
static int curcol,ccv;

/* Umrechnung vertikale Pos. -> Wert 0-15 */
USHORT VertPos(g)
struct Gadget *g;
{
struct PropInfo *pi;
long p;
  pi=(struct PropInfo *)g->SpecialInfo;
```

Listing: Colreq.c

```
   p=67715L-(long)(pi->VertPot);
   p/=4369;
   if (p>15) p=15;
   return(p);
}

/* Setzen der Farbe & Eintragen in StrGadget */
ChgCol(n,w,m,z)
int n,w;
USHORT m;
int z;
{
   if (z>-1) {  /* Mehr als ein Register ? */
      ccv&=m;
      ccv|=(w<<((2-z)<<2));
   }
   else ccv=w;
   SetRGB4(vp,(long)n,(long)ccv>>8,(long)((ccv>>4)&0xF),(long)(ccv&0xf));
   ITOA(&COLTXT[z],w);
   RefreshGList(&COLTXTG,win,NULL,1L);
}

int ColReq(pal,cols)
USHORT *pal;  /* Zeiger auf Palette */
int cols;   /* Anzahl Bitplanes */
{
int x,y,x1,y1,n=1,t,r;

long class;
int code;

struct IntuiMessage *msg,*GetMsg();

struct Gadget *gad;
int gid;
VOID *AllocMem();

   /* Alles vorhanden was gebraucht wird ? */
   if (!(IntuitionBase&&GfxBase&&(cols>0)&&(cols<5))) return(-1);

   scrdef.Depth=cols;
   if (!(scr=OpenScreen(&scrdef))) return(-1);

   windef.Screen=scr;
   if (!(win=OpenWindow(&windef))) {CleanUp();return(-1);}

   rp=win->RPort;
   vp=&(scr->ViewPort);

   LoadRGB4(vp,pal,(long)(1<<cols));

   /* Original-Palette kopieren */
   for (t=0;t<(1<<cols);t++) Palette[t]=pal[t];

   SetDrMd(rp,JAM2);

   x=sx[cols];
   y=sy[cols];
   n<<=cols;

   /* Farb-Rechtecke zeichnen */
   for (x1=0;x1<64;x1+=x) {
      for (y1=0;y1<64;y1+=y) {
         SetAPen(rp,(long)((1<<cols)-n));
         RectFill(rp,(long)(81+x1),(long)(13+y1),
               (long)(80+x1+x),(long)(12+y1+y));
         n--;
      }
   }

   /* Screen reinschieben */
   for (x=0;x<19;x++)
      MoveScreen(scr,0L,-8L),
      WaitTOF();

   /* Default-Farbe: 1 */
   ColorDisp(1,pal);

   for (r=0;r==0;) {
      WaitPort(win->UserPort);
      msg=GetMsg(win->UserPort);

      while (msg) {

         class=msg->Class;
         code=msg->Code;

         switch (class) {

            case CLOSEWINDOW:
               /* Abbruch, Palette restaurieren */
               for (t=0;t<(1<<cols);t++) pal[t]=Palette[t];
               r=2;
               break;

            case GADGETUP:
               gad=(struct Gadget *)msg->IAddress;
               gid=gad->GadgetID;
```



Listing: Colreq.c

```c

        switch (gid) {

          case 20:  /* Abbruch, alte Palette */
            for (t=0;t<(1<<cols);t++) pal[t]=Palette
  [t];
            LoadRGB4(vp,pal,(long)(1<<cols));
            x=curcol;
            curcol=-1;
            ColorDisp(x,pal);
            break;

          case 21:  /* OK, neue Palette */
            r=1;
            pal[curcol]=ccv;
            break;

          case 14:
            /* String-Gadget Eingabe */
            x=strlen(COLTXT);
            x1=y=0;

            /* Hex-String -> Integer */
            while (x>0) {
              if ((COLTXT[x1]>='0')&&
                (COLTXT[x1]<='9'))
                y1=COLTXT[x1]-'0';
              else {
                y1=toupper(COLTXT[x1])-'A'+10;
                if (y1<0) y1=0;
                if (y1>15) y1=15;
              }
              y<<=4;
              y+=y1;
              x--;
              x1++;
            }

            ChgCol(curcol,y,0,-1);
            SetProps(y);
            break;

          case 10:
            /* Neue Farbe auswaehlen */
            x=msg->MouseX-81;
            y=msg->MouseY-13;

            x/=sx[cols];
            y/=sy[cols];

            ColorDisp(x*fx[cols]+y,pal);
            break;

          case 11:
            /* R - Gadget */
            x=VertPos(gad);
            ChgCol(curcol,x,0x0FF,0);
            break;

          case 12:
            /* G - Gadget */
            x=VertPos(gad);
            ChgCol(curcol,x,0xF0F,1);
            break;

          case 13:
            /* B - Gadget */
            x=VertPos(gad);
            ChgCol(curcol,x,0xFF0,2);
            break;
        }
        break;
      }

      ReplyMsg(msg);
      msg=GetMsg(win->UserPort);
    }
  }

  /* Screen wegschieben */
  for (x=0;x<19;x++)
    MoveScreen(scr,0L,8L),
    WaitTOF();

  /* Aufraeumen */
  CleanUp();
  return(r);
}

CleanUp()
{
  if (win) CloseWindow(win);
  if (scr) CloseScreen(scr);
}

/* Masken */
#define RED    0xF00
#define GREEN  0x0F0
#define BLUE  0x00F

ColorDisp(n,pal)
int n;
USHORT *pal;
{
struct RastPort *rp;
int color,r,g,b;

  if (curcol!=n) { /* Neue Farbe angewaehlt ? */

    if (curcol>=0)
      pal[curcol]=ccv;
    rp=&(scr->RastPort);

    if (n==0) {  /* Hintergrund-Farbe ? */
      SetAPen(rp,1L);
      RectFill(rp,30L,15L,130L,65L);
      SetAPen(rp,0L);
      RectFill(rp,32L,16L,128L,64L);
    }
    else {
      SetAPen(rp,(long)n);
      RectFill(rp,30L,15L,130L,65L);
    }

    curcol=n;
    ccv=color=pal[n];
    SetProps(color);
  }
}

/* Integer nach Hex-Ascii, ein Nibble */
ITOA(c,w)
char *c;
USHORT w;
{
  if (w>15) w=15;
  if (w<10)
    *c=w+'0';
  else
    *c=w-10+'A';
}

/* Proportional-Gadgets setzen,
   String Gadget aktualisieren */
SetProps(color)
int color;
{
int r,g,b;
  r=(color&RED)>>8;
  g=(color&GREEN)>>4;
  b=(color&BLUE);

  NewModifyProp(&RGADG,win,NULL,
    AUTOKNOB|FREEVERT,-1L,(long)(65535-r*4369),
    -1L,4096L,1L);

  NewModifyProp(&GGADG,win,NULL,
    AUTOKNOB|FREEVERT,-1L,(long)(65535-g*4369),
    -1L,4096L,1L);

  NewModifyProp(&BGADG,win,NULL,
    AUTOKNOB|FREEVERT,-1L,(long)(65535-b*4369),
    -1L,4096L,1L);

  ITOA(&COLTXT[0],r);
  ITOA(&COLTXT[1],g);
  ITOA(&COLTXT[2],b);

  RefreshGList(&COLTXTG,win,NULL,1L);
}
/*********** Ende von Colreq.c **********************/
```

Listing: Colreq.c

```c
/*****************************************************/
/*                    ColTest.c                      */
/*         Programm zum Testen von `ColReq'          */
/*       (C) 1990 by G. Glendown & AmigaDOS          */
/*             Sprache: Aztec C                      */
/*****************************************************/

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
VOID *OpenLibrary();

USHORT Palette[] = {
  0x000,0xfff,0x222,0x555,0x888,0xBBB,0xEEE,0xF00,
  0x0F0,0x00F,0xF0F,0xFF0,0x0FF,0xF80,0xF08,0x80F};



main()
{
int t;
  IntuitionBase=OpenLibrary("intuition.library",0L);
  GfxBase=OpenLibrary("graphics.library",0L);
  t=ColReq(Palette,4);
  t=ColReq(&Palette,3);
  t=ColReq(&Palette,2);
  t=ColReq(&Palette,1);
  if (IntuitionBase) CloseLibrary(IntuitionBase);
  if (GfxBase) CloseLibrary(GfxBase);
}
```

Listing: Coltest.c



Liebe Leser,
in dieser Rubrik finden Sie neben Informationen zu den in der AMIGA DOS vorgestellten Programmen und Produkten Rat und Hilfestellung zu Ihren kleineren und größeren Programmierproblemen. Wir sind natürlich jederzeit bemüht, die eingehenden Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für die anderen in unserer Zeitschrift beantwortet.
Dringende Probleme, die das Heft betreffen, lassen sich möglicherweise besser telefonisch regeln. Rufen Sie dienstags unsere Hotline an. Von 17 bis 20 Uhr stehen wir Ihnen mit Rat und Tat zur Seite, wenn Sie eine der folgenden Nummern wählen:
0 56 51/8 09-7 40 (bis 744) *Ihre AMIGA-DOS-Redaktion*

SIE FRAGEN, WIR ANTWORTEN

Hier nun wieder unsere Leserbriefecke, in der Sie Fragen stellen oder beantworten können. Wenn Sie uns schreiben wollen, richten Sie Ihr Schreiben an den

DMV-Verlag,
Redaktion AMIGA DOS,
Leserservice,
Postfach 250,
3440 Eschwege.

Virus, Virus...

Als absoluter Anfänger auf dem Amiga habe ich noch wenig Ahnung und möchte mich mit einer Frage an Sie wenden. Ich möchte gerne das Programm Virusx in das C-Verzeichnis meiner Workbench kopieren, um neu eingeladene Disketten auf Viren zu untersuchen. Leider habe ich nicht die geringste Ahnung, wie das vonstatten geht.
Eine zweite Frage: Habe ich es richtig verstanden, daß ein Virus, gleich welcher Art, nur durch verseuchte Disketten in den Computer gelangen kann, so daß man davon ausgehen kann, daß der Computer bei ausschließlicher Benutzung von Orginalsoftware von Viren verschont bleibt? Ich gehe davon aus, daß der Computer nicht mit einem anderen Rechner verbunden ist. Daran schließt sich gleich eine neue Frage an. Wie seriös sind die Anbieter von PD-Disketten, und muß man grundsätzlich damit rechnen, virenverseuchte Disketten angedreht zu bekommen?

Jochen Stude, Goldbach

Um Virusx in das C-Verzeichnis zu kopieren, geht man folgendermaßen vor:

- Workbench laden und Shell (oder CLI) öffnen
- »Copy ?« eingeben und [Return] drücken
- nach dem Laden die Diskette einlegen, auf der sich Virusx befindet
- eingeben: »df0:virusx (richtigen Namen einsetzen) to ram:«
- warten, bis das Programm ins RAM kopiert wurde
- Workbench einlegen
- eingeben:»copy ram:virusx (richtigen Namen eingeben!) to df0:c«
- warten, bis kopiert wurde.

Um Virusx sinnvoll zu nutzen, sollte man jetzt mit dem Editor ED die Startup-Sequenz laden (»ed s/startup-sequenz«) und als erste Zeile eingeben »c:virusx« (auch hier richtigen Namen einsetzen). Virusx wird so beim Start der Workbench automatisch geladen.
Zu den Viren: Ihre Informationen sind richtig. Viren verbreiten sich primär durch Disketten. Der Ablauf dabei ist folgender: Ein Virus hat eine Diskette infiziert. Die Diskette wird benutzt und der Virus gelangt in den Speicher des Amiga. Solange der Computer nun nicht ausgeschaltet wird, vervielfältigt sich der Virus, indem er Kopien von sich selbst auf jede neu eingelegte Diskette schreibt. Natürlich ist dieser Ablauf extrem vereinfacht dargestellt.
PD-Anbieter sind bemüht, ihren Kunden 'saubere' Ware zu verkaufen, und arbeiten daher ebenfalls mit Antivirus-Programmen. Sollten Sie es doch mal mit einer verseuchten Diskette zu tun bekommen, ist es am besten, den Hersteller zu informieren und eine neue Disk zu verlangen.

(Red.)

Festplattenschalter

Ich habe einen Amiga 2000B mit einer Festplatte und möchte gerne wissen, ob man an der Stromleitung zur Harddisk einen Schalter anbringen kann, um die Festplatte auszuschalten (bei Spielen beispielsweise brauche ich die Platte nicht). Nur weiß ich nicht, welches Kabel ich durchtrennen muß, um den Schalter dazwischenzulöten.

Marcel Würzburg, Offenburg

Einfach einen Schalter an die Stromversorgung zu löten, geht bei Festplatten nicht, denn hier gibt es 'jemanden' der etwas dagegen hat – der Controller. Autoboot-Festplatten und ihre Controller lassen sich nur durch anliegende Signalinformationen oder durch Festplattenhilfsprogramme abschalten.

(Red.)

Eingabehilfe

Seit Anfang des Jahres beziehe ich die Zeitschrift AMIGA DOS. Leider habe ich keine Eingabehilfe, um die Listings aus den Heften einzugeben. Bitte senden Sie mir diese, oder teilen Sie mir mit, welche ich benutzen kann.

Günter Jock, Hürth

In Ausgabe 1/90 finden Sie die Programme »CheckBas« für Amiga-BASIC-Programme und »HexType« für Hexdump-Listings.

(Red.)

Amiga 500 als PC

Inzwischen halte ich die vierte Ausgabe von AMIGA DOS in den Händen und muß sagen, daß dieses Magazin alles bisherige weit in den Schatten stellt. Mit großem Interesse verfolge ich den leicht verständlichen Assembler-Kurs, wirklich ein gelungenes Werk. Wie wäre es denn, nur so als Anregung, mit einem C-Kurs für Anfänger? Das mit dem A68K-Assembler klappt ja schon recht gut bei mir, aber Assembler finde ich ein bißchen langwierig und unübersichtlich.
So, nun eine Frage, die mich brennend interessiert: Von ei-

nem Bekannten hörte ich, daß es für den Amiga 500 eine PC-Einsteckkarte von der Firma SCS Schomburg (oder so ähnlich) gibt. Existiert eine solche Karte tatsächlich? Wenn ja, gibt es dazu auch einen umfassenden Hardwaretest und in welchen himmlischen Dimensionen bewegt sich der Preis?
Oder gibt es irgendeine andere Möglichkeit, PC-Programme (speziell Turbo Pascal) auf dem Amiga 500 laufen zu lassen?

Stefan Korn
Aalen

Uns ist bekannt, daß es demnächst eine PC-Karte für den A500 von der Firma Roßmöller geben wird (wir werden noch genauer darüber informieren), die von Ihnen genannte ist uns zur Zeit noch nicht bekannt.
Ein C-Kurs wird wohl ab Heft 10 oder 11/90 beginnen.

(Red.)

Fragenkatalog

Als begeisterter Leser Ihrer Zeitschrift bin ich sehr erfreut über den von Ihnen angebotenen Service, auf die Fragen der Leser zu antworten.
Meine erste Frage betrifft das Thema Viren: Kann sich ein Virus allein beim Einlegen einer infizierten Diskette, zum Beispiel auf der Workbench oder bei Hilfsprogrammen wie dem Climate oder Diskmaster in den Computer installieren?
Weiterhin interessiert es mich, ob es eine Möglichkeit gibt, die Reaktion der linken Maustaste zu verbessern, da diese Aussetzer beim Betrieb zeigt, beziehungsweise erst auf stärkeren Druck reagiert.
Von Bekannten habe ich gehört, daß es ratsam sei, regelmäßig eine Reinigungsdiskette zu benutzen. Stimmt das, und wenn ja, wie oft soll ich reinigen?
Ich besitze einen Amiga 2000 mit eingebauter zweiter Floppy. Allerdings stört mich das nervenaufreibende Rattern der zweiten Floppy, wenn keine Diskette eingelegt ist. Gibt es eine Möglichkeit dieses Rattern (softwaremäßig) abzustellen? Als ich die zweite Floppy eingebaut habe, fiel mir auf, daß entgegen den Angaben in Büchern die zweite RAM-Platine, auf der sich eigentlich die zweiten 512 kBytes des RAM-Speichers befinden sollten, fehlt. Da ich meinen Computer im Januar 89 gekauft haben, ist es wohl unmöglich, daß ich den neuen Fat-AGNUS-Chip im Computer habe. Jedoch laufen Programme, die ein MByte benötigen, einwandfrei. Wo ist die Platine?
Hat es irgendwelche Auswirkungen auf ein Programm, wenn ich eine benötigte Library wie zum Beispiel Icon-Library, mit einem Packer-Programm behandele?
Kann man bei GFA-BASIC 3.0 mit denselben Befehlen arbeiten wie bei Amiga-BASIC 1.2, oder ist ein größeres Umlernen von Befehlen erforderlich?
Da ich mit dem Gedanken spiele, mir eine PC-Karte zuzulegen, möchte ich gerne wissen, ob es für alle MS-DOS-Programme genügt, daß der Amiga die Bildschirmdarstellung und die Maus emuliert. Zum Abschluß noch eine letzte Frage: Überall wird wie selbstverständlich von »Source-Code« oder »Quell-Code« geredet. Was ist das denn überhaupt?
Wo gehobelt wird, fallen auch Späne, wird im allgemeinen gesagt. So ist auch bei Ihrer Zeitschrift ein wenig Kritik anzuwenden. Der BASIC-Kurs ist meiner Meinung nach für den totalen Anfänger, den Sie damit wohl ansprechen wollen, eher schwer zu verstehen. So kann man es wohl als etwas unglücklichen Versuch bezeichnen, wenn unbekannte Befehle verwendet werden, um dem Laien zu erklären, daß in Amiga-BASIC keine Zeilennummern gebraucht werden. Ich finde, daß der Anwender dadurch nur verwirrt wird. Ihre Idee, Aufgaben an das Ende jedes Kursteils zu stellen, finde ich ausgezeichnet, doch auch hier werden wieder unbekannte Befehle verwendet, die nur kurz und flüchtig erklärt werden. Auch dies ist nur verwirrend.

Oliver Fischer,
Duisburg

Uff, viele Fragen auf einmal. Versuchen wir es trotzdem:
1) Zum Thema Viren siehe Brief Nr. 1.
2) An der Maus etwas zu verbessern, ist nicht ohne weiteres möglich. Einzig der Einbau von Mikroschaltern wäre zu überlegen, da rentiert sich doch eher der Einsatz einer neuen Maus (Opto-Maus, siehe Bericht im Heft, oder Reisware-Maus).
3) Eine Reinigungsdiskette kann ab und zu (bei uns jedes halbe Jahr) benutzt werden, sie sollte jedoch für Naß/Trokkenreinigung sein, also ein weiches Vlies mit aufgeträufelter Reinigungsflüssigkeit enthalten.
4) Uns ist nicht bekannt, wie man das Rattern abstellen kann.
5) Die zweiten 512 kByte befinden sich ebenfalls auf der Hauptplatine.
6) Systemprogramme und Libraries sollte nicht 'gecruncht' werden, da sie nicht mehr als zum System gehörig erkannt werden.
7) GFA-BASIC und Amiga-BASIC unterscheiden sich in einigen Belangen ganz erheblich.
8) In der Grundausstattung (also 'nur' XT- oder AT-Karte) laufen auf dem Amiga nur Programme, die den CGA (Color-Graphics-Adapter)-Modus haben. Hercules, EGA- und VGA-Programme brauchen die entsprechenden Grafikkarten.
9) Source-Code oder Quelltext ist der Grundtext eines Compiler- oder Assembler-Programms. Dieser Text ist meist reiner ASCII-Code und kann mit Editoren oder Textverarbeitungen erstellt werden.

(Red.)

Assemblerkurs

Erst vor kurzem entdeckte ich Ihren Assemblerkurs, der mich sehr interessiert. Ich habe auch schon seit längerem den A68K-Assembler, aber leider keine Bedienungsanleitung dazu. So weiß ich zum Beispiel nicht, was die mit einem Backslash eingeleiteten Buchstaben bedeuten. Das auf derselben Diskette befindliche Doc-File gibt nur begrenzt Hilfe. Wo kann ich eine genauere Beschreibung des Assemblers bekommen?

Walter Kramer,
Klagenfurt

Ob es eine deutsche Anleitung zum A68K gibt, ist uns nicht bekannt, wir werden jedoch in einer der nächsten Ausgaben genauer auf den PD-Assembler eingehen.

(Red.)

Rahmenlose Fenster

Mit meinem Amiga 2000 habe ich folgendes Problem: In unregelmäßigen Abständen verschwindet beim Booten der PC-Erweiterung der 'Fensterrahmen' »PC - Color Display«. Der PC läßt sich dann immer noch einwandfrei bedienen, und nach Betätigung der rechten Maustaste erscheint nach wie vor das Menü. Meine Frage ist nun, wie man den 'Fensterrahmen' dauerhaft beseitigt, beziehungsweise wie man generell Amiga-Fenster ohne Rahmen programmiert.

Frodo von Beublinger,
Mühlheim

Bei der Window-Struktur läßt sich das »GimmeZeroZero«-Bit setzen, es erlaubt den Aufbau rahmenloser Windows. Näheres über Window- und Screen-Strukturen finden Sie in Büchern über System- und Intuition-Programmierung.

(Red.)

Kann nicht lesen, kann nicht schreiben...

Ich habe ein kleines Problem mit meinem Amiga 500 und hoffe, auf diesem Wege Hilfe zu bekommen. Jedesmal, wenn ich vom CLI (WB 1.2) aus mehrere Dateien, zum Beispiel ein Direktory, zu kopieren versuche, belohnt das Betriebssystem meine Versuche mit Read-/Write-Errors ohne Ende. Benutze ich die Version 1.3, ist es das gleiche Spiel. Sogar die neue Kickstart 1.3 schafft keine Abhilfe. Ein Virus kann es eigentlich auch nicht sein, da der Fehler schon auftritt, solange ich den Rechner besitze. Außerdem blieben die 1001 Virenkiller, die ich an die Workbench herangelassen habe, ohne Erfolg. Ein zweites Laufwerk brachte auch keine Verbesserung des Problems. Benutze ich hingegen ein Kopierprogramm, erhalte ich 100% fehlerfreie Kopien. Ansonsten wollte ich noch sagen, daß mir die AMIGA DOS gut gefällt, weil aus allen Bereichen rund um den Amiga berichtet wird. Nicht ganz so gut hat mir der Artikel über die Reisware-Maus gefallen, wo unter anderem steht: "Was das Innenleben anbelangt, unterscheiden sich beide Mäuse nur unwesentlich voneinander." Dem kann ich nicht ganz zustimmen, da bei der Reisware-Maus Kunststoffrollen anstelle der verchromten Eisenrollen der Orginalmaus verwendet wurden. Der Vorteil dabei ist nämlich, daß beim Reinigen der Maus die Chromschicht, sobald Sie einmal verletzt ist, abblättert und die Eisenrollen sofort zu

rosten anfangen. Dadurch wird eine feinfühlige Steuerung der Maus unmöglich. Dies konnte ich nicht nur an meiner eigenen Maus feststellen, es stellte sich auch bei mehreren Bekannten ein.
Zum Schluß würde ich gerne noch wissen, ob die Programme, die in den Heften abgedruckt sind, auch auf Disketten erhältlich sind.

Ingo Lowinski,
Aachen

Alle Programme, die in der AMIGA DOS erscheinen, sind auf der mit dem Heft erscheinenden DATABOX erhältlich, sowohl in lauffähiger (wenn möglich) Form, wie auch als Quelltexte.
Die von Ihnen genannten Read-/Write-Errors lassen sich eigentlich nur durch einen defekten oder verstellten Schreib-/Lesekopf erklären. In diesem Fall wäre eine Reparatur angebracht.

(Red.)

Wer, Was, Wann, Wo...

B. Haidner erkundigte sich in der Ausgabe 5/90 auf der Leserbriefseite nach einem Programm, das ihn nach dem Einschalten an wichtige Daten und Termine erinnert.
Sie verweisen auf das Programm 'Wer, Was, Wann, Wo'. Zu einem weitaus günstigeren Preis bietet das Public Domain Programm »Nag«, das sich auf der Fred-Fish-Disk 161 befindet, die von Herrn Haidner gewünschte Funktion. Insbesondere läßt es sich in die Startup-Sequenz einfügen und ist so direkt nach dem Booten des Rechners verfügbar.

Hartmut Schumacher,
Berlin

Vielen Dank für den Tip.

(Red.)

Kritik und Lob

Seit März lese ich Ihre Zeitschrift sehr aufmerksam durch. Im Vergleich zu anderen Amiga-spezifischen Zeitschriften ist mir die AMIGA DOS positiv aufgefallen. Dazu muß ich Ihnen mitteilen, daß ich zur Zeit Student bin und meinen 500er haupsächlich für Anwendungen und seltener für Spiele benutze. Daher interessieren mich am stärksten die Hardware-Erweiterungen wie RAM-Karten, Festplatten und Drucker, sowie Textverarbeitung, DTP, Utilities, C, Basic und mathematische Programme. Wie Sie lesen können, war ich bisher noch wenig kreativ (Musik, Malen) tätig.
Gut finde ich Ihr Verhältnis zum Computer. Sie sehen die Möglichkeiten der Maschine kritisch und nicht als ein großes Wunderwerk der Technik. In anderen Magazinen habe ich das Gefühl, daß die Autoren vor überschwenglichem Lob vergessen haben, daß der Amiga zwar außergewöhnliche Fähigkeiten besitzt, aber in vielen Bereichen von der Professionalität noch recht weit entfernt ist (Software-Problem). Er ist eben als Homecomputer ausgelegt. Mir gefällt die Darstellung der Themen. Meistens wird ein Artikel fast naiv begonnen.
So ist der Einsteiger über die Grundlagen informiert, und der 'alte Hase' ist eher amüsiert als gelangweilt. Am Ende fühle ich mich eigentlich immer gut informiert. Aufgefallen ist mir auch, daß Sie meine Kreativität sehr stark herausfordern. So habe ich nach den Kursen, der Werkstatt und nach den kleinen Hardwarebasteleien immer große Lust, mich gleich an das Gerät zu setzen. So bin ich richtig traurig, daß ich in diesen Monaten meinen 500er nicht dabei habe und so meine Ideen zurückstellen muß. Übrigens, Ihrem negativen Eindruck von dem Buch 'Amiga Hardwaretuning' muß ich, soweit ich es gelesen habe, beipflichten.
Der große Teil mit der deutschen PD ist mir sehr wichtig. Es ist wohl der beste Weg, das 'Schwarzkopieren' ein wenig einzudämmen. Ihre Beschränkung auf den deutschen Markt ist akzeptabel, da dieser in letzter Zeit sehr vielfältig geworden ist.
Negativ ist mir im letzten Heft aufgefallen, daß die Listings im Vergleich zum Heftumfang und zu den Erläuterungen einen großen Raum einnehmen. Das soll doch nicht auf eine Werbung für Ihre Databox hinauslaufen?
Die Spiele lassen sich meistens in ein bestimmtes Schema einpassen und bieten selten etwas neues. Daher sollte dieser Teil sich nicht vergrößern und sich eher auf besondere Produkte fokussieren und andere wie gewohnt nur kurz vorstellen. Übrigens, von Lösungen für Adventures halte ich nichts und lese sie daher auch nicht.
Dem Magazin fehlt noch der aktuelle Bezug, was wahrscheinlich an der noch nicht abgeschlossenen Startphase liegt. So würde ich gerne etwas über die Cebit und die interne PC-Karte von Roßmöller erfahren. Einige Fragen sind mir auch noch geblieben. Zu 'Amiga im Eigenbau': Benötigt der Computer ein spezielles Gehäuse? Oder kann man auch eines aus Holz oder sogar Plexiglas selbst bauen? Lassen sich Filecards an den Amiga 500 anschließen und wenn ja, welche? Wann können externe (RAM, Harddisk) und interne (RAM, PC-Karte) Erweiterungen nicht miteinander laufen?
Wodurch sind Begrenzungen für die Größe des internen Speichers gegeben (AGNUS)?

Ralf Friedrichsen
Weil am Rhein

Es freut uns natürlich, wenn wir Ihren Geschmack getroffen haben, aber auch Ihre Kritiken verhallen nicht ungehört. Natürlich ist ein relativ neues Heft wie die AMIGA DOS noch nicht das Supermagazin, das allen gerecht werden kann, aber wir bemühen uns und meinen, daß wir uns auf dem richtigen Weg befinden. Für Anregungen und Kritiken haben wir jedenfalls immer ein offenes Ohr.
Zu den Fragen:
Natürlich können Sie sich Ihr Gehäuse selbst basteln, es bedeutet nur mehr Arbeit. Bei Preisen ab 70,– DM für fertige C-Gehäuse ist dies aber auch schon wieder eine Extra-Überlegung wert.
Filecards an den 500er anzuschließen, ist im Augenblick noch Bastelarbeit, da unseres Wissens noch keine Slots für den Amiga 500 existieren (aber auch hier lassen wir uns gerne belehren).
Probleme gibt es meistens, wenn beide Geräte autokonfigurierend sind und die gleiche Konfigurationsbelegung benutzen. Eines der beiden Geräte ist damit nicht mehr 'vorhanden'.
Der Prozessor kann maximal 16 MByte adressieren, durch interne Verdrahtung sind beim 500er Speichererweiterungen bis 8 MByte möglich.
Der Fat AGNUS kann dabei nur 512 kByte benutzen, der Big AGNUS dagegen schon 1 MByte.
Listings im Heft sollen natürlich qualitativ hochwertig sein, müssen also schon etwas für die Abtipparbeit bieten. Mehr Qualität bedeutet allerdings auch mehr Quantität, wir bemühen uns jedoch, die Abtipperei für Sie so erträglich wie möglich zu machen.

(Red.)

Die Problemecke

An dieser Stelle finden Sie Fragen, die redaktionell nicht oder kaum zu klären sind und die wir daher an die Leser der AMIGA DOS weitergeben.

Zuviel Pink im Druck

Seit einem viertel Jahr besitze ich den Farb-Matrixdrucker Star LC-10 Color und habe mit diesem folgende Probleme. Eine ausgedruckte Grafik weist immer (mal mehr, mal weniger) Farbfehler auf, die sich in mehreren, bis jetzt nur pinkfarbenen Balken äußern. Ferner sind einige Teile der Grafik viel schwächer gedruckt als andere. Ausgedruckt wurden die Bilder mit dem Zeichenprogramm D-Paint III, aber auch bei Versuchen mit anderen Programmen erhielt ich stets das gleiche Resultat.

Dan Andruschow
Berlin

Dies scheint eines der leidigen Druckertreiber-Probleme zu sein. Wir konnten hier keine Lösung finden. Sollten Sie, liebe Leser, etwas darüber wissen, schreiben Sie uns bitte.

AMIGA DOS -TIP – die aktuelle Produktanzeige

Commodore® Ersatzteil Service

✖ Wir liefern für **Händler** und Privatanwender preiswert und prompt

✖ Rufen Sie uns an: (02331-43001) oder schreiben Sie uns:

CIK-Computertechnik • Ingo Klepsch
Berliner Straße 49b • D-5800 Hagen 7
TELEFAX: 02331-42499



Tino Hofstede
Computerservice
An der Windmühle 8
5010 Bergheim 5



WARRIORS, SF-Adventure	19,90
HOTELMASTER, Spiel	19,90
ENGLISCHTRAINER incl. Vokabeln	39,90
FRANZÖSICHTRAINER incl. Vokabeln	39,90
VIDEOVERWALTUNG	49,90
DISK-ETIKETTEN-Druck/Programm	19,90
LIGAVERWALTUNG (6-20 Mannschaften)	49,90
PD-DISKETTEN	ab 2,60
KATALOG (Schutzgebühr in Briefmarken)	1,00

FÜR JEDES TEIL EINE ANDERE QUELLE? – BEI SCHÄFER IST VIEL AN EINER STELLE!

AMIGA FLACHBETT A4 SCANNER **DM 948,-**
Kann als BILDERFASSUNGSGERÄT/Kopierer und Thermodrucker eingesetzt werden. Scanndichte 200 Punkte/Zoll, Scannzeit 10 Sekunden. Ablage des GANZEN Bildes im IFF, Auflösungen 320 x 200, 640 x 400, 320 x 256, 640 x 512 werden unterstützt. Ausschnittvergrößerungen möglich. Binär + 16 Grau-Darst.

VIDEO TEXT-DECODER **DM 298,-**
Man kann das VIDEOSIGNAL eines Recorders, Scart TV oder Tuners dazu verwenden, den freien Service des Teletextes im IFF- oder ASCII-Format abzulegen. Super-Grafik-Darstellung.

MINI MAX 500 **DM 288,-**
Erweiterung bis 2MB, 512 K bestückt. Preise für Aufrüstsätze siehe GIGATRON-Inserat.

AMIGA-ACTION-REPLAY **DM 189,-**
Modul für den Erweiterungssteckplatz des Amiga 500. Funktionen: Freezen (Programme aus dem Computerspeicher auf Diskette abspeichern); Spieletrainer (unbegrenzte Leben); Spriteeditor (Sprites sichtbar machen und verändern); Virusdetector (Viruserkennung); Bilder und Musik auf Diskette in IFF-Format abspeichern zur Weiterverarbeitung, Programm-Neustart (nach Unterbrechung); Computer-Status (Fast-RAM, Chip-RAM, RAM-Disk, Floppy-Status usw) u.v.m.

ZWEITLAUFWERK 3,5" (KONYO-SENATOR)	**239,-**
MIDI-MASTER	**99,-**
MAXON-JUNIORPROMMER (2716-27011)	**249,-**
BURST-NIBBLER (Syncro-Expreß) mit Hardware	**149,-**
AMIGA-DEVPACK-ASSEMBLER (M&T)	**149,-**
MUSIC MANAGER	**49,-**
MIDI MASTER + MUSIC MANAGER	**120,-**
SOUND SAMPLER	**169,-**

alle Bücher von Markt & Technik – umfangreiche Software und Ersatzteile auf Anfrage – Preisliste DM 2,50 in Briefmarken – Anrechnung bei Kauf – Versand nur gegen Vorkasse + 5,00 oder Nachnahme + 8,00 DM.

(CLS) – COMPUTERLADEN SCHAEFER
Klingelholl 111, 5600 Wuppertal 2, Tel.: 02 02/50 81 21
Geschäftszeiten: Mo+Di, Do+Fr 14-18.30 Uhr, Sa 10-13 (14) Uhr

---

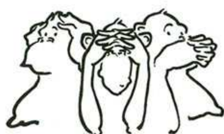
Ihre Hotline für AMIGA-DOS-TIP

Tel.: 089/4391087 bis -89
Fax: 089/4391080

R + S Computer

GFA-BASIC 3.5 Interpreter	198,-
GFA-BASIC Compiler	99,-
GFA-ASSEMBLER	139,-
GFA-ZOETROPE	179,-
Strategiespiele in GFA-BASIC	99,-

R + S Computer
Am Tömp 46
4050 Mönchengladbach 4
Tel.: 02166 / 54840



TROPENWALD- VERNICHTUNG, WIESO? IST DOCH SO WEIT WEG.

Irrtum. Der tropische Regenwald reguliert auch unser Klima.

Der tropische Regenwald wird gnadenlos vernichtet. Jedes Jahr eine Fläche, die fast so groß wie die Bundesrepublik Deutschland ist. Mit katastrophalen Folgen für das Weltklima. Aber auch für die im Tropenwald lebenden Ureinwohner und Tier- und Pflanzenarten.

Wenn Sie wissen wollen, wie der tropische Regenwald gerettet werden kann, dann füllen Sie den Coupon aus. Wir beantworten Ihre Frage, sagen Ihnen, wo wir uns noch engagieren, warum Sie uns dabei helfen können.

Ja, ich will von ROBIN WOOD wissen, wie der tropische Regenwald gerettet werden kann.

Name, Vorname

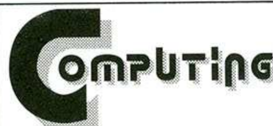
Straße

Wohnort

Weil ich helfen will, lege ich 3 Mark in Briefmarken bei.
Coupon bitte einsenden an
ROBIN WOOD, Postfach 10 21 22, 2800 Bremen 1

dmv

ROBIN WOOD
Gewaltfreie Aktionsgemeinschaft für Natur und Umwelt e.V.



1180 Wien, Schulgasse 63
Tel: 0222/408 52 56 Fax: 0222/408 99 78
1100 Wien, Gudrunstraße 158
Tel: 0222/602 26 18
Postversand - Teilzahlung - Leasing

Amiga 500 Speichererweiterung 512 KB	öS 1.590,–	(DM 227,-)
Amiga 3.5 Zoll Laufwerk, Bus/Ein- u. Ausschalter	öS 1.790,–	(DM 256,-)
Amiga 5.25 Zoll Laufwerk, Bus/Ein- u. Ausschalter, 40/80 Tr.	öS 2.490,–	(DM 356,-)
Amiga 2000, 2 MB Memory	öS 5.990,–	(DM 856,-)
Amiga 500 mit 80 MB SCSI Harddisk	öS 16.990,–	(DM 2427,-)
Control-Center 500 (Top-Styling!)	öS 1.490,–	(DM 213,-)
Audiomaster II	öS 990,–	(DM 142,-)
Hires Workbench	öS 349,–	(DM 50,-)
CLI-Manager	öS 299,–	(DM 43,-)
Digi View 4.0	öS 2.490,–	(DM 356,-)
Elan-Performer	öS 1.190,–	(DM 170,-)
Genlock Pal Ver 1.3	öS 4.990,–	(DM 713,-)

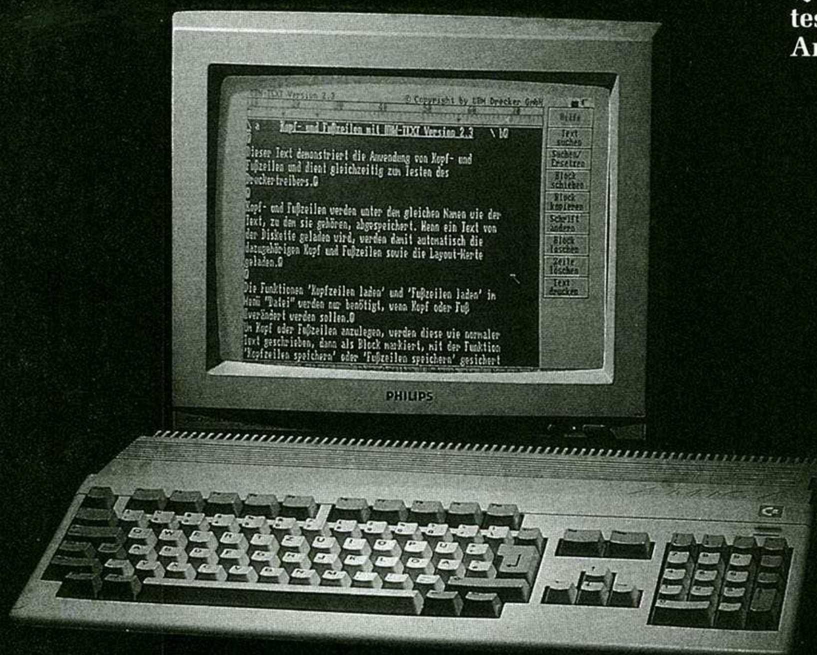
Eurosystems	(Midi-Manager, Pro Sampler, Syncro Express, Handyscan)	lagernd
GVP	(45 MB 28 ms, 40 MB 19 ms, 80 MB 19 ms, 68030 Karte,...)	lagernd

Alle Preise inkl. 20 % MWSt., Druckfehler und Preisänderungen vorbehalten.

Qual der Wahl

Textverarbeitung auf dem Amiga

Wer die Wahl hat, hat zugegebenerweise auch die Qual. Wo im MS-DOS-Bereich bereits ein sehr breites Spektrum an Textverarbeitungen vorliegt, ist die Anzahl für den Amiga noch gering.



Seit rund fünf Jahren ist der Amiga ein Dauerbrenner auf dem Markt. In diesem Zeitraum ist die Zahl der Textverarbeitungen ständig gestiegen. Was die Qualität der einzelnen Textverarbeitungsprogramme hinsichtlich des Preis-Leistungsverhältnisses anbelangt, klaffen jedoch hier und da recht große Lücken. Waren anfangs Umlaute und Sonderzeichen noch Fremdwörter für eine Textverarbeitung, so gehört beispielsweise die WYSIWYG-Darstellung ("Watt ju Sii Is Watt ju Gätt" zu deutsch: die Darstellung auf dem Bildschirm entspricht dem späteren Ausdruck) bei vielen Programmen bereits zum Leistungsumfang.

Die Anfänge

Blättern wir zurück: Die Zeiten, als Dichter und Denker ihre Texte noch mit Tinte und Federkiel niederschrieben, sind längst Geschichte. Auch Kugelschreiber oder Füllfederhalter gehören der Vergangenheit an. Selbst die Arbeit an der Schreibmaschine gestaltete sich anfangs noch recht mühsam. Zunächst waren es rein mechanische Schreibmaschinen, auf denen Sekretärinnen ihre tägliche Post erledigten oder Schriftsteller ihre geistigen Ergüsse niederschrieben. Doch schon bald nahm die Entwicklung ihren Lauf. Die mechanischen Schreibmaschinen wurden durch die elektrischen ersetzt.

Mit dem Einsatz von Computern trat jedoch eine revolutionäre Wende ein. In der heutigen Zeit haben diese Computer unter Verwendung eines Textverarbeitungsprogramms in der Vielzahl der Betriebe, Büros und an den heimischen Schreibtischen die Arbeit der Texterstellung übernommen. Der Vorteil gegenüber den herkömmlichen Schreibmaschinen liegt klar auf der Hand: Zum einen können sie Texte speichern, der Speicherplatz ist nur abhängig von der Größe der Speichermedien, zum anderen sind die Texte so jederzeit greifbar, bearbeitbar und lassen sich ebenso leicht ausdrucken. Kurzum, die Schrecken nahezu jeder Vorzimmerdame, einen Brief zum hundertsten Mal in die Schreibmaschine zu tippen, gehören dank Computer und Textverarbeitung der Vergangenheit an.

Mit den geeigneten Programmen ist diese Arbeit fast zu einem Kinderspiel geworden. Text eingeben, speichern, ausdrucken, bei Korrekturen wieder neu einladen – das alles stellt bei einer halbwegs guten Textverarbeitung kein Problem mehr dar. Selbst um Tippfehler muß man sich kaum noch Gedanken machen, da in vielen Programmen ein sogenannter Spell-Checker enthaltem ist. Doch muß man solche Leistungen natürlich auch entsprechend bezahlen.

Orientierungshilfen...

sind gefragt, damit die Qual der Wahl etwas leichter fällt. Woran soll sich der Käufer orientieren, wenn er sich eine Textverarbeitung zulegen möchte? Nach welchen Kriterien kann die richtige Auswahl getroffen werden? Kaum ein Händler hat die ganze Palette vorrätig, die es auf dem Markt zu kaufen gibt. Daher fehlen auch jegliche Vergleichsmöglichkeiten zwischen den einzelnen Produkten; man hat meist nicht die Möglichkeit, die verschiedenen Programme eingehend selbst zu prüfen. 'Prüf vor Kauf' wäre eigentlich die richtige Methode, um später nicht enttäuscht zu werden, wenn das Programm die Wünsche nicht erfüllt. Doch dies ist meist nicht möglich. Was also tun? Wir wollen Ihnen so gut es geht Entscheidungshilfen geben.

Welche Arbeit soll das Programm leisten? Möchte man es in erster Linie zum professionellen Einsatz nutzen und legt außerdem noch großen Wert auf die Druckqualität, sollte die Wahl in Richtung DTP-Programme (Desktop-Publishing) zielen. Da der Aufwand bei der Erstellung eines DTP-Dokumentes mit einem hohen grafischen Anteil sehr hoch ist, wird die Arbeitsgeschwindigkeit dieser Programme auch entsprechend langsamer (wenn man nicht in der glücklichen Lage ist, über eine Turbokarte zu verfügen). Andererseits sprechen aber wiederum die Ergebnisse, sprich der Ausdruck, für die Verwendung dieser Programme im professionellen beziehungsweise semiprofessionellen Bereich. Zählt man sich zu dem großen Heer der Vielschreiber, empfiehlt es sich, eine Textverarbeitung zu wählen, mit der sich in erster Linie Texte bearbeiten lassen, deren Arbeitstempo relativ hoch ist.

Der Vielschreiber

Der charakteristische Vielschreiber legt nicht unbedingt Wert auf eine möglichst große Vielzahl unterschiedlicher Schriften und Fonts, sondern er möchte möglichst schnell, sicher und unkompliziert seine schriftliche Arbeit bewerkstelligen. Sehr von Vorteil wäre in diesem Fall auch eine automatische Sicherung des Geschriebenen.

Daneben sollte die Bedienbarkeit entsprechend einfach und unkompliziert sein. Hier würde sich beispielsweise ein Programm anbieten, das sich sowohl mit der Maus, als auch von der Tastatur aus steuern läßt. Eine lexikalische Überprüfung des erstellten Textes würde sich ebenfalls anbieten, jedoch sollte die Überprüfung sinnigerweise auch in deutscher Sprache stattfinden. In der Regel liegt solchen Lexika ein fester Basiswortschatz zugrunde.

Eine andere Möglichkeit bietet der sogenannte Thesaurus, ein Synonymlexikon, das dem Vielschreiber nützliche Dienste erweisen kann, indem beispielsweise sinnverwandte Wörter zur Auswahl vorgeschlagen werden. Hat man es bei seiner Vielschreiberei in erster Linie mit wissenschaftlichen Texten zu tun, sollte das Programm nach Möglichkeit über eine automatische Fußnotenverwaltung verfügen, damit ein mühseliges, nachträgliches Einfügen dieser Fußnoten entfällt. Zusammenfassend sollte die Textverarbeitung für den Vielschreiber wie folgt aussehen: Es sollte ein Programm sein, bei dem man sofort problemlos losschreiben kann, ohne erst unbändige Handbücher wälzen zu müssen.

Der "Ab-und-zu-Schreiber"

Unter den Amiga-Benutzern gibt es sicherlich eine sehr große Zahl, die sich in vorderster Front mit dem Schreiben von Programmen beschäftigen und die dazu keine Textverarbeitung, sondern einen Editor der entsprechenden Programmiersprache verwenden. Zusammen mit den Anwendern, die ihren Amiga ab und an einmal zum Briefeschreiben benutzen, benötigt diese Gruppe kein sündhaft teures Programmpaket, sondern hier reicht eigentlich ein kleines, kompaktes Textverarbeitungsprogramm vollständig aus. An dieser Stelle wäre ein wenig mehr bereits zuviel, denn zusätzliche Funktionen oder eine Vielzahl spezieller Optionen würden den Anwender nur verwirren. Zudem wäre dieses bißchen natürlich entsprechend teurer in der Anschaffung. Eines sollte jedoch bedacht werden: Beabsichtigt man in naher Zukunft, sich doch etwas mehr mit dem Schreiben von Texten zu beschäftigen, sollte man nicht auf jeden Pfennig schauen, sondern eine Textverarbeitung wählen, die auch höheren Ansprüchen genügt.

Sicherlich läßt sich nicht jeder in diese beiden Kategorien eingliedern. Sie soll auch nur ein grobes Raster darstellen und zwei Hauptgruppen unterscheiden. Jeder Anwender hat bei der Auswahl "seiner" Textverarbeitung gewisse Bedürfnisse, die er an das Programm stellt. Wir haben daher neun unterschiedliche Textverarbeitungsprogramme ausgewählt und für Sie getestet. Unter diesen Programmen ist sicherlich für jeden Anwendertyp etwas dabei.

Einige Grundlagen

Wir haben die Textverarbeitungsprogramme nach folgenden Kriterien getestet:

- Kurzinfo
- Bedienung
- Leistungsmerkmale
- Besonderheiten

Sie können die einzelnen Ergebnisse der Tests der Marktübersicht "Textverarbeitungen" entnehmen. Dort finden Sie die wichtigsten Kriterien stickpunktartig aufgelistet.

Ein großer Vorteil dieser Programme gegenüber anderen Softwareprodukten ist die Tatsache, daß Textverarbeitungsprogramme in den seltensten Fällen über einen Kopierschutz verfügen. Es lassen sich also auch ohne Probleme Sicherheitskopien

anlegen und die Originale können geschont werden. Die Hersteller bieten nach erfolgter Registrierung verschiedene Dienstleistungen an. Mit dem Kauf eines Originalproduktes ist beispielsweise oft ein Update-Service verbunden, das heißt, ein registrierter Benutzer erhält entweder kostenlos oder zu einem geringen Entgelt die jeweils neueste Version des Produktes geliefert. Viele Hersteller sind auch dazu übergegangen, einen Telefonservice anzubieten. Dort werden Probleme, die der Anwender mit dem Programm hat, sofort an Ort und Stelle von Experten gelöst.

Ein weiteres Problem sollte im Zusammenhang mit Textverarbeitungsprogrammen nicht verschwiegen werden. Möchte man wirklich den gesamten Funktionsumfang seines Programmes nutzen, stellt sich die Frage nach dem Speicherbedarf. Bei einigen der getesteten Programme liegt der Speicherplatzbedarf bei mindestens 1 MByte, der überwiegende Teil gibt sich mit 512 kByte zufrieden. Doch stellt diese Angabe eigentlich einen Trugschluß dar, denn die Funktionsfülle der meisten Programme macht einen größeren Speicherbedarf erforderlich. Möchte man also in den wirklichen Genuß seiner Textverarbeitung kommen – sprich alle vorhandenen Features auch nutzen – sollte man sich unbedingt eine Speichererweiterung von mindestens 1 MByte zulegen, um ein sinnvolles Arbeiten zu gewährleisten.

Kriterien für den Leistungsumfang

Die Diskettenverwaltung innerhalb eines Textverarbeitungsprogramms sollte klar und übersichtlich gelöst sein.

Das bedeutet, daß hier auch ein eigenes Fenster vorhanden sein muß. Innerhalb dieses Fensters sollte für jedes angeschlossene Laufwerk auch ein Icon zur Verfügung stehen, um einen möglichst problemlosen Laufwerkswechsel vollziehen zu können. Ebenso wäre es von Vorteil, wenn zwischen Verzeichnisangabe und Dateinamen ohne große Umstände gewechselt werden kann. Demgegenüber ist eine individuelle Textkennung durch Anhängen einer programmspezifischen Endung nicht unbedingt notwendig. Sie dient zwar der Übersicht, (Dateien werden vorsortiert) schließt aber die Darstellung der Dateien, die über diese Endung nicht verfügen, aus.

Sichern um jeden Preis

Texte, die sich im Arbeitsspeicher befinden, sind durch eventuelle Systemabstürze dauernd in Gefahr. Deshalb ist ein Zwischenspeichern eigentlich unumgänglich. Wer hat nicht selbst schon die Tragik erlebt (auch unsere Redaktion blieb davon nicht verschont): Im Haus wird gebaut. Man hat einen Text zu zwei Dritteln fertiggestellt und dann passiert das Schreckliche – der Strom fällt aus, und man hat vergessen, den Text abzuspeichern. Die ganze Arbeit für die Katz. Hier wäre eine regelmäßige, automatische Sicherung des Textes vorteilhaft. Besonders die Vielschreiber dürften diese Problematik kennen. Leider verfügt nur ein Produkt über dieses Feature. Für den "Otto Normalverbraucher" reicht es vollkommen aus, wenn eine Sicherheitskopie des entsprechenden Files angelegt wird. Diese Kopie wird dann unter einem anderen abgespeichert, meist durch die Endung .BAK (Backup-Datei) gekennzeichnet. Auf diese Datei kann dann im Notfall zurückgegriffen werden.

Handlich sollte ein Handbuch schon sein, jedoch sollte es sich nicht auf diese Eigenschaft allein beschränken. Man kann fast jedes Programm über den simplen Trial-and-Error-Weg erkunden, jedoch ist diese Art, die Feinheiten kennenzulernen, oft sehr mühselig. Ein Manko besteht darin, daß manche Textverarbeitungsprogramme aus der englischsprachigen Welt stammen. Der Umgang mit einem entsprechenden Handbuch setzt dann notgedrungen die Kenntnis der englischen Sprache voraus. Hier hadern viele Anwender doch etwas mit dem Schicksal, denn der Umgang mit Handbüchern kann sehr schnell zum Alptraum werden. Einige Firmen sind bereits dazu übergegangen, die Handbücher ins Deutsche zu übersetzen, doch sind auch hier noch einige Mängel zu erkennen, weil allzu wörtlich aus dem Englischen übernommen wird und der Übersetzer oftmals wenig Kenntnis der Materie besitzt. Wie hat nun ein gutes Handbuch auszusehen?

Die Handbücher

Es sollte dem Anwender und besonders dem Einsteiger die wichtigsten Informationen über das Programm in leicht verständlicher und übersichtlicher Form vermitteln. Dazu gehören unter anderem Informationen zu wichtigen Punkten wie Installation des Programms, ein Stichwortverzeichnis zum schnellen Nachschlagen oder ein umfangreicher Referenzteil.

Was ist sinnvoller, eine reine Mauskontrolle oder die Bedienung über die Tastatur? Im Zweifelsfall ist es günstig, wenn man beide Möglichkeiten hat. Nahezu jedes getestete Programm verfügt über ein Pull-Down-Menü, das sich einfach mit der Maus bedienen läßt.

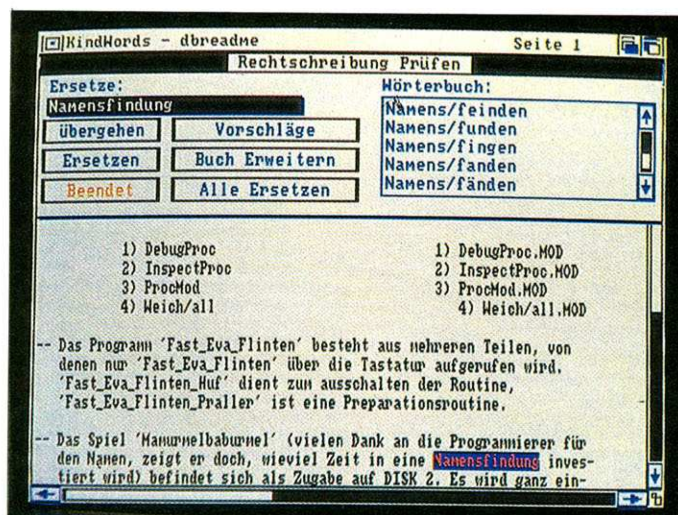


Bild 1. Sinn oder Unsinn eines deutschsprachigen Lexikons

Komfortables Arbeiten

Dazu fährt man auf den entsprechenden Befehl und hält dabei die rechte Maustaste gedrückt. Danach führt der Amiga den entsprechenden Befehl aus. Diese Handhabung ist relativ langsam. Demgegenüber verfügen sehr viele Programme außerdem über sogenannte Shortcuts (zu deutsch: Abkürzungen), Tastenkombinationen, die durch das gleichzeitige Drücken zweier Tasten (zum Beispiel der Amiga- und einer anderen Taste) ebenfalls den entsprechenden Befehl ausführen. Sehr sinnvoll ist die Arbeit mit einem sogenannten Clipboard. Wie mit der Schere läßt sich ein Textstück ausschneiden und in einen Puffer schieben, wo er bis auf weiteres verbleibt und bei Bedarf wieder hervorgeholt werden kann.

Die Funktion Suchen & Ersetzen darf ebenfalls in keiner Textverarbeitung fehlen, die den Anspruch erhebt, komfortabel zu sein. Word Wrapping ist in den meisten Programmen vorhanden, das heißt, paßt ein Wort nicht mehr in die Zeile, wird es automatisch ganz in die nächste Zeile mitübernommen. Die automatische Silbentrennung wäre hier das geeignetere Hilfsmittel, um Texte relativ unkompliziert bearbeiten zu können. Bei der Frage, ob ein Programm über WYSIWYG verfügen soll oder nicht, scheiden sich die Geister. Denn jeder Anwender hat seine besonderen, individuellen Wünsche, gerade auch was die Arbeitsgeschwindigkeit anbelangt. Hier muß klar festgehalten werden, daß eine höhere Arbeitsgeschwindigkeit fast immer auf Kosten der Darstellungsweise eines Textes geht. Man sollte daher sein Urteil über die Leistungsfähigkeit eines Programms nicht unbedingt von dem Fakt abhängig machen, ob es über WYSIWYG verfügt oder nicht.

Etwas fürs Auge – Grafikeinbindung

Will man's gern nett aufbereiten, sollte das Programm über die notwendigen Möglichkeiten der Einbindung von Grafik

verfügen. Das bedeutet, es muß in der Lage sein, ein Standardformat auch einlesen zu können. Hier sind sicherlich DTP-Programme im Vorteil, jedoch gibt es bei den getesteten Programmen einige recht gute Lösungen. Sie erreichen zwar nicht die Qualität von entsprechenden Malprogrammen, sondern ändern meist die Farben in Graustufen um oder reduzieren die Anzahl der Farben. Aber wie bereits erwähnt, wünscht man eine professionelle Lösung, sollte die Zielrichtung ein DTP-Programm sein.

Wie sinnvoll sind Zusatzprogramme?

Eine andere Form, den Text aufzubereiten, ist die Verwendung verschiedener Schriftarten oder Fonts. Auch hier stellen die Programme meist ein reichhaltiges Angebot zur Verfügung. Zusätzlich besteht die Möglichkeit, andere Fonts einzuladen. Die Public Domain stellt hier eine große Palette an

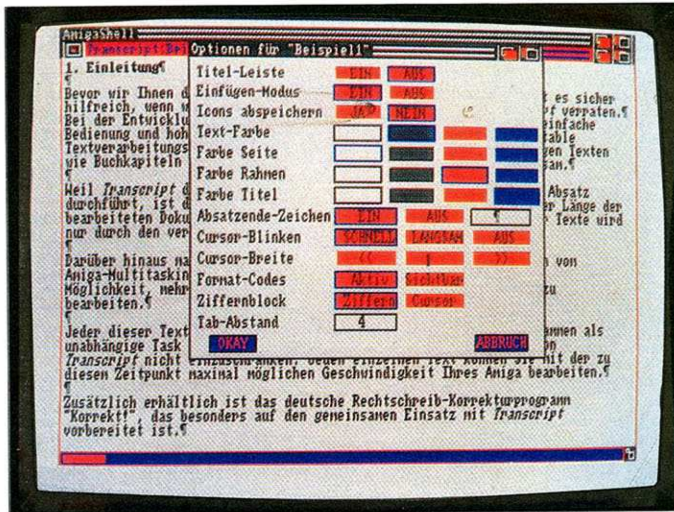


Bild 2. Transcript – der Sprinter

Schriftarten und Fonts zur Verfügung.

Der Preis einer Textverarbeitung hängt sehr oft davon ab, welche zusätzlichen Programme zur Verfügung gestellt werden. Hier muß der Anwender dann aber die Frage nach Sinn und Unsinn zusätzlicher Produkte stellen. Was ist unbedingt notwendig, um effektiv arbeiten zu können? Die Vielschreiber unter den Lesern wären sicherlich für eine direkte Rechtschreibkorrektur dankbar. Doch die Realität setzt hier recht schnell die Grenzen. Besonders die deutsche Sprache ist durch einen sehr reichhaltigen Wortschatz mit zahlreichen Synonymen gekennzeichnet. Schaut man sich die Kompendien diverser Lexika an, wird sehr schnell klar, daß es eigentlich schier unmöglich scheint, ein kompaktes Lexikon zur Rechtschreibprüfung zu erstellen.

In diesem Zusammenhang muß auch auf einen weiteren Punkt hingewiesen werden. Importierte Programme, deren Ursprung im englischsprachigen Raum anzusiedeln ist, haben hier die schlechtesten Karten. Denn was nützt einem das beste Lexikon, wenn der Text in deutscher, die Rechtschreibprüfung jedoch in englischer Sprache vollzogen wird. Bild 1 macht deutlich, was bei einer extremen Übersetzung herauskommen kann, wohlgemerkt einer 'extremen'. Demgegenüber gibt es den sogenannten Thesaurus, ein Zusatzprogramm, das dem Anwender Hilfe verspricht. Dieses Programm ermöglicht es, sinn- und sachverwandte Wörter anzuzeigen. Damit hat der Vielschreiber ein recht nützliches Hilfsmittel an der Hand. Über dieses Zusatzprogramm verfügen lediglich Wordperfect und Excellence, die Programme,

MÄUSE

NACHT FÜR NACHT WURDE DIE MAUS RÜDIGER VON DEMSELBEN SCHRECKLICHEN ALPTRAUM HEIMGESUCHT ...

© Stiller 90

die sich auch preislich deutlich von den übrigen Testkandidaten absetzen.
Recht sinnvoll erscheinen jedoch Zusatzprogramme, die sich mit dem Umwandeln und Aufbereiten von Grafik beziehungsweise Bildmaterial beschäftigen oder einfach weitere Schriften und Fonts zur Verfügung stellen. Was die einzelnen Programme an Leistungen bringen, soll im folgenden etwas näher beleuchtet werden. Es wird dabei jedoch kein Anspruch auf Vollständigkeit erhoben, es sollen in erster Linie Orientierungshilfen gegeben werden.

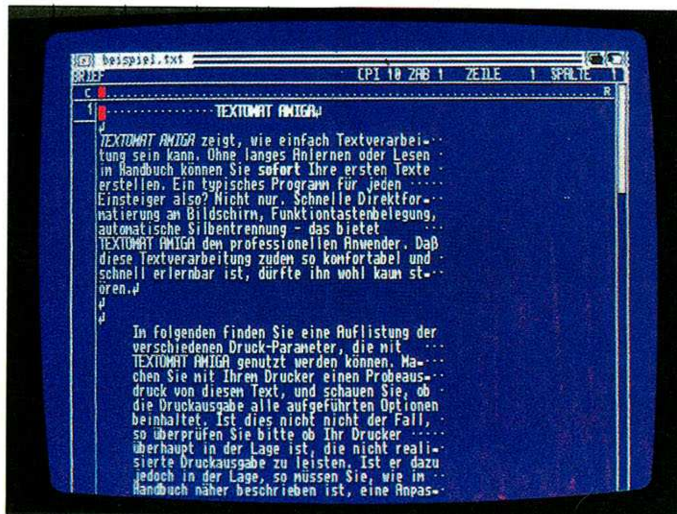


Bild 3. Textomat – schmaler Beckertext

Transcript

Wir haben für Sie die deutsche Version von Transcript getestet. Dieses Programm zeichnet sich durch ein schnelles, unkompliziertes Arbeiten aus. Natürlich hat hier die Geschwindigkeit auch ihren Preis, denn diese Schnelligkeit geht auf Kosten der Grafik. Dafür ist die Arbeitsgeschwindigkeit des Programms nicht unbedingt von der Länge des zu bearbeitenden Textes abhängig. Hier setzt lediglich der Speicherplatz, der dem Benutzer zur Verfügung steht, eine Grenze. Die Bedienerführung ist sehr übersichtlich gehalten. Das Programm läßt sich mit der Maus und über Shortcuts bedienen. Die Dateiverwaltung ist sehr übersichtlich gestaltet, für jedes Laufwerk ist ein entsprechendes Icon vorhanden, das angeklickt werden kann. Transcript erlaubt die Definition von Macros und natürlich deren Verwendung. Die Multitaskingfähigkeiten des Amiga kommen voll zum Tragen. Das Programm ist in der Lage, mehrere Texte in verschiedenen Fenstern zu bearbeiten, wobei für jeden Text auch eine eigene Task läuft. Neben einer Serienbrieffunktion verfügt das Programm über die Möglichkeit, ein Index- oder Stichwortverzeichnis zu erstellen.

Ein weiterer Vorteil von Transcript ist die Tatsache, daß es mit dem DTP-Programm Professional Page ohne Probleme zusammenarbeitet. Dazu dienen eigene Format-Codes, die Transcript für die Textformatierung bereitstellt. Beide Programme ergänzen sich recht sinnvoll. Das deutsche Handbuch ist übersichtlich gegliedert und mit einem Index versehen, in dem sich wesentliche Suchbegriffe schnell nachschlagen lassen.

Die Erläuterungen sind leicht verständlich, so daß Einsteiger kaum Probleme haben dürften. Zusätzlich kann das Rechtschreibkorrekturprogramm Korrekt erworben werden. Anwender, die relativ viel schreiben, sind mit diesem Programm gut bedient.

Textomat

Mit diesem recht einfach gehaltenen Programm lassen sich in erster Linie Texte relativ schnell bearbeiten. Die Dateiverwaltung ist übersichtlich gehalten. Bei der Dateiauswahl erwartet Textomat als Suffix .TXT, es lassen sich natürlich auch Texte mit anderem Suffix einladen. Die wichtigsten Befehle und Funktionen können per Maus und auch über Shortcuts erreicht werden. Textomat erlaubt die Belegung der Funktionstasten. Das Programm verfügt ferner über die Möglichkeit, Texte nicht nur den verschiedenen Laufwerken zuzuleiten, sondern auch den seriellen Port anzusteuern. Mit dem Programm lassen sich Bilder, die im IFF-Format gespeichert wurden, in den Text einladen. Auch der Textmodus läßt sich auf 12x12 verändern (standardmäßig liegt er bei 8x8). Die gängigsten Schriftarten stehen ebenso zur Verfügung wie die automatische Silbentrennung. Im Programmrepertoire sind auch die geläufigsten Druckertreiber vertreten, so daß der Ausdruck der Texte keine großen Probleme bereiten dürfte. Textomat stellt sich als abgespeckte Version des Programms Beckertext dar, bei der jedoch einige wichtige Features fehlen. Dazu zählt unter anderem die Rechenfunktion. Dies wird natürlich auch in dem niedrigeren Preis von Textomat gegenüber dem großen Bruder Beckertext deutlich. Zusätzlich läßt sich der Upgrade-Service von Beckertext nutzen. Aufgrund einiger Mängel (wie beispielsweise Indexerstellung) ist dieses Programm für den Vielschreiber nicht unbedingt geeignet. Jedoch ist der Anwender, dessen Tätigkeitsfeld sich in erster Linie auf das Schreiben von Briefen erstreckt, mit Textomat gut bedient.

Beckertext V1.02

Kommen wir zum großen Bruder von Textomat. Dieses Programm stellt sich bereits etwas komfortabler vor. Die Ladefunktion klappt ein Fenster auf, bei der man die Wahl zwischen sechs verschiedenen Menüpunkten hat. Die Dateiverwaltung entspricht der von Textomat und ist dementsprechend übersichtlich. Die



Bild 4. Beckertext – der große Bruder

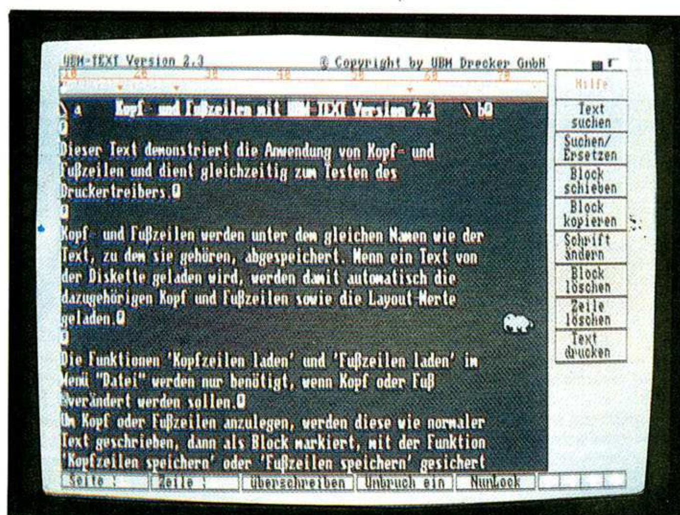


Bild 5. UBM-Text – die Übersichtliche

WYSIWYG-Darstellung ist in der vorliegenden Version 1.02 zwar vorhanden, soll jedoch in der neuen Version 2.0 über weitaus mehr Features verfügen. Mit Beckertext kann nicht nur ein einzelner Text bearbeitet werden, sondern das Programm nutzt auch die Multitaskingfähigkeiten des Amiga. Je mehr Seiten man allerdings pro Text und Fenster benötigt, desto schneller schränkt sich die Anzahl der zu öffnenden Fenster auch ein.

Grafiken können im IFF-Format eingeladen werden. Die Umsetzung läßt wie bei Textomat doch etwas zu wünschen übrig. Bei farbigen Bildern streikt das Programm schlichtweg, das heißt, sie werden lediglich in schwarzweiß dargestellt. Das integrierte Lexikon ermöglicht in der Online-Funktion ein sofortiges Überprüfen des gerade Geschriebenen. Der Text kann in der Form bearbeitet werden, daß ein Teil in einen extra vorgesehenen Speicher gepackt und bei Bedarf wieder aus diesem Speicher ausgelesen wird. Daneben verfügt Beckertext über die Möglichkeit, ein Index- oder Stichwortverzeichnis anzulegen.

Dies ist für Vielschreiber ein wichtiges Utensil. Jedoch sind dem Programm weder Macros noch Fußnoten bekannt. Gegenüber Textomat verfügt das Programm über neun verschiedene Druckerfonts, die angesteuert werden können. Ein sehr positiver Aspekt ist jedoch noch nicht erwähnt worden. Mit Beckertext kann man auch Berechnungen anstellen wie beispielsweise eine Spaltenberechnung.

Die Bedienung ist über Maus und auch über Shortcuts möglich und erlaubt so ein recht schnelles Arbeiten. Das deutsche Handbuch ist sehr übersichtlich gestaltet und wird in einem Ringordner geliefert. Die Erläuterungen sind sehr umfangreich und auch für den Einsteiger gut verständlich. Daneben verfügt das Handbuch über ein Stichwortverzeichnis und diverse Anhänge wie Druckeranpassung etc.

Der Preis hält sich für ein Textverarbeitungsprogramm in erträglichem Rahmen. Für Vielschreiber ist Beckertext in der vorliegenden Form bedingt einsetzbar. Dem Briefeschreiber steht aber ein kompaktes und gutes Programm zur Verfügung.

UBM-Text V2.3

Mit diesem Programm stellt sich eine Textverarbeitung vor, die bereits auf den ersten Blick sehr kompakt erscheint. Sehr viele Funktionen sind standardmäßig am Bildschirm präsent (wie zum Beispiel Zeilenlineal, Seiten- und Zeilenanzeige, Help etc.). Sie lassen sich aber auch mit dem Menüpunkt Hilfsanzeigen entweder teilweise oder ganz an- und ausschalten. Die Dateiauswahlbox zeigt direkt das aktuelle Laufwerk an, von dem das Programm gestartet wurde. Das Programm erwartet als Suffix .UTXT, ist aber auch in der Lage, über den Menüpunkt ASCII-Datei, Texte mit anderem Suffix einzuladen und zu bearbeiten. UBM-Text kann entweder per Maus oder mit den Shortcuts bedient werden. Dabei sind neben den Amiga-Kombinationen (Amiga-Taste + normale Taste) auch Kombinationen mit den Ctrl-Tasten möglich. Oft benutzte Redewendungen lassen sich in einem Puffer ablegen und bei Bedarf über die Funktionstasten F5 bis F10 wieder abrufen. Ein weiterer Puffer wird für das sogenannte Clipboard zur Verfügung gestellt, in das Blöcke abgelegt und wieder entnommen werden können. Das Druckermenü erlaubt die Ansteuerung der Schnittstellen PAR:, PRT: und SER:, daneben stehen noch zwei Zeichensätze zur Verfügung. Das Erstellen von Serienbriefen ist mit UBM-Text relativ unkompliziert, ebenso sind diverse Anpassungen (Drucker oder Tastatur) recht problemlos. Die Einbindung von farbiger Grafik im IFF-Format ist möglich, jedoch muß das entsprechende Bild zuvor mit dem Zusatzprogramm Graph-Cut bearbeitet werden. Dieses Programm gehört zum Lieferumfang wie auch das Programm UBM-Druck und diverse Druckertreiber und bereitet die Bilder UBM-Text-entsprechend auf. Diese Bilder kann man an der Endung .UPIC erkennen, andere Formate werden nicht eingeladen. Das Programm arbeitet in der Bildschirmdarstellung nach dem WYSIWYG-Verfahren, dabei ist die Verarbeitungsgeschwindigkeit noch relativ hoch. Das Handbuch

ist übersichtlich und auch für Einsteiger leicht verständlich. Es verfügt daneben über diverse Anhänge und ein umfangreiches Stichwortverzeichnis. Das Programm eignet sich sehr gut, um Briefe grafisch nett aufzubereiten, jedoch weniger für akademische Arbeiten. Das heißt, für den Vielschreiber ist es weniger geeignet.

ProWrite

Dieses Textverarbeitungsprogramm ist wie viele andere Programme in der Lage, Grafiken in den Text einzubauen. Dieses gestalterische Hilfsmittel ist zudem sehr flexibel zu handhaben, selbst der Text an sich kann beispielsweise mit Farbattributen versehen werden. Die Bedienung ist auch hier wie bei vielen anderen Produkten über Maus und Shortcuts recht handlich. Das vorhandene Rechtschreiblexikon ist wie das gesamte Programm in englischer Sprache, was so manchen Anwender vielleicht abschrecken könnte. Die Überprüfung kann direkt während der Eingabe ablaufen (Online-Funktion). Bei den auszuwählenden Punkten ist der Menüpunkt Charakter recht umfangreich, es stehen neben den normalen Schriftarten noch weitere Fonts zur Gestaltung des Textes zur Verfügung. Die Dateiauswahlbox ist im Vergleich zu den anderen Testkandidaten etwas gewöhnungsbedürftig. Das Handbuch fällt etwas mager aus, verfügt aber trotzdem über Anhang, Glossar und Index. Man sollte der englischen Sprache doch schon etwas mächtig sein, um eventuellen Übersetzungsmißgeschicken aus dem Weg zu gehen. Fehlende Features wie Fußnotenverwaltung, Indexverwaltung oder Macros lassen eigentlich nur den Schluß zu, daß dieses Programm wohl für Briefeschreiber, jedoch nicht für Vielschreiber angeraten ist.

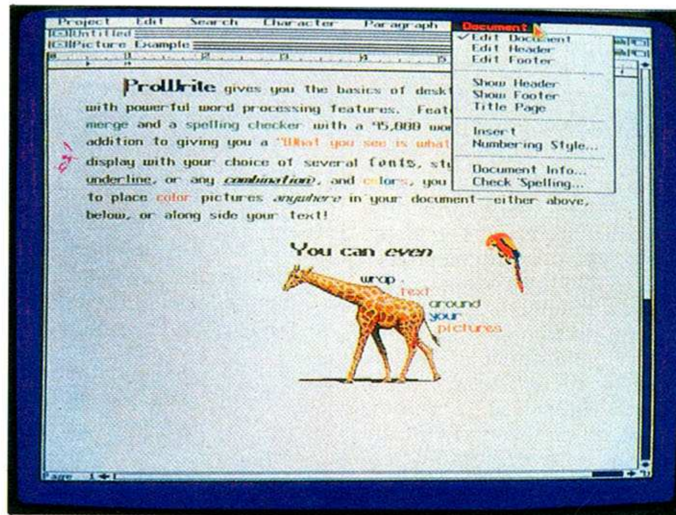


Bild 6. Prowrite – die Flexible

Vizawrite Desktop 2.02

Diese Textverarbeitung gehört zu den Programmen, bei der die WYSIWYG-Darstellung zu den positiven Aspekten zählt. Formatierungsbefehle, der Einsatz von Kopf- und Fußzeilen erscheint beim Druck so, wie er bereits auf dem Bildschirm angezeigt wird. Im Programmpaket ist eine zusätzliche Diskette mit den verschiedensten Schriftarten und Fonts enthalten, um dem Anspruch, ein Desktop-Programm zu sein, gerecht zu werden. Dieser Anspruch wird auch durch die Tatsache unterstrichen, daß Vizawrite in der Lage ist, Bilder in den Text einzuladen.

Dabei ist es egal, mit welchem Programm die Bilder erstellt wurden. Es können auch eingescannte oder digitalisierte Bilder eingelesen werden. Voraussetzung ist lediglich, daß die Bilder im IFF-Standardformat abgespeichert sind. Da Vizawrite in den vier Farben der Workbench arbeitet, ist es nicht in der Lage, Farbbilder entsprechend darzustellen, diese werden in acht verschiedenen Grauwerten gezeichnet. Für die Arbeit im semiprofessionellen Bereich eignet sich dieses Programm durchaus. Die Bedienung ist sowohl per Maus als auch über Shortcuts recht einfach zu vollziehen. Angenehm ist die Arbeit mit Textbausteinen, in denen sich häufig verwendete Floskeln oder anderes speichern lassen und jederzeit in das aktuelle Dokument eingeladen werden kann. Daneben verfügt Vizawrite über eine Serienbrieffunktion. Die Dateiauswahlbox ist übersichtlich strukturiert und zeigt das jeweils aktuelle Laufwerk an. Für einen allumfassenden Einsatz fehlen jedoch so notwendige Features wie Fußnotenverwaltung oder Indexverwaltung. Das Handbuch ist relativ knapp gehalten und könnte hier und da ein wenig mehr Erläuterung einzelner Befehle und Funktionen vertragen. Es fehlt beispielsweise ein Stichwortverzeichnis zum schnellen Nachschlagen. Wer seine Briefe, Rundschreiben oder auch Werbeschriften etwas aufpeppen möchte, ist mit diesem Programm sehr gut bedient. Wer jedoch sein Hauptaugenmerk auf die Arbeit mit wissenschaftlichen Texten legt, dem fehlen hilfreiche Features wie Fußnoten etc.

Word Perfect

Mit diesem Programm stellt sich eine Textverarbeitung vor, die vornehmlich durch ihr breit gefächertes Leistungsspektrum besticht. Neben den obligatorischen Grundfunktionen jeder komfortablen Textverarbeitung findet der Anwender eine Reihe von weitergehenden Features. So wenden sich Funktionen wie Macrodefinition, Blockfunktionen, Fußnoten-

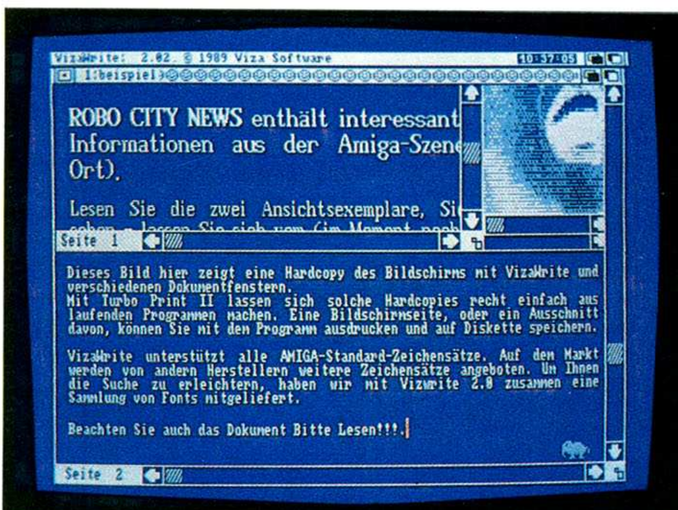


Bild 7. Vizawrite – Desktop im Schnellverfahren

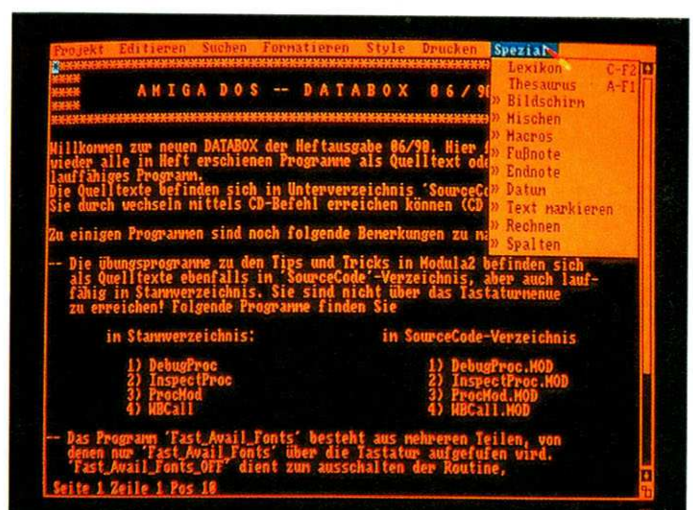


Bild 8. Word Perfect – das Kraftpaket

verwaltung, Lexikonfunktion und mehrspaltige Formatierung an den Vielschreiber, während Optionen wie zum Beispiel Schriftstücke mischen das Programm auch bei kleineren Texterfassungsarbeiten interessant machen. Jedoch wird der Preis von rund 400 DM Privatanwender wohl eher verschrecken. Allerdings erhält der Benutzer für den stolzen Preis neben der eigentlichen Textverarbeitung auch ein rund 600seitiges, gut strukturiertes deutsches Handbuch.

Ein Kundendienst, der auch telefonisch zur Verfügung steht, unterstreicht den professionellen Charakter des Programms. Word Perfekt beinhaltet auch eine Lexikonfunktion, beläßt es allerdings nicht dabei, sondern geht noch einen Schritt weiter: Thesaurus überprüft Texte auf Wortwiederholungen durch und ersetzt mehrfach verwendete Wörter nach Belieben durch Wörter aus einem breiten Stichwortverzeichnis. Dieses Verzeichnis enthält zirka 415000 Einträge und kann vom Anwender modifiziert werden. Auf das Einbinden von Bildern und vergleichbaren DTP-Features muß der Word-Perfekt-Benutzer verzichten. Word Perfekt ist sicherlich eines der am weitesten entwickelten Programme seiner Art auf dem Amiga und wird wegen seiner professionellen Aufmachung besonders das Interesse von Vielschreibern wecken. Negativ macht sich die umständliche Bedienung des Produkts bemerkbar, so wird die Maus im Programmablauf fast gar nicht genutzt. Wort Perfect setzt hier auf Tastaturkommandos, was aber wieder dem Vielschreiber durch die schnelle Ausführbarkeit entgegenkommt. Das Programm unterstützt mehr als 200 verschiedene Druckertypen und läßt sich auf Festplatte installieren. Wer sich durch den hohen Preis nicht schocken läßt, der erhält eine Textverarbeitung, die wenige Wünsche offenläßt.

KindWords

Die Textverarbeitung 'KindWords' gehört von ihrer Entwicklung her zu den Zwittern unter den getesten Programmen. Das Herkunftsland unterliegt den englischsprachigen Gepflogenheiten, die Textverarbeitung ist an deutsche Textschreiber angepaßt. Dementsprechend sind einige Funktionen 'gemischt europäisch' zu nennen, vor allem, was die enthaltene Rechtschreibprüfung anbelangt. Daß man den guten alten Amiga mit 'Magda' korrigieren kann, reizt immer noch zu Lachanfällen. Abgesehen davon ist KindWords ein 'stilles Wasser' – und die sind bekanntlich tief. KindWords besitzt alles, was eine gute Textverarbeitung ausmacht. Serienbriefe, Blockoperationen, Zeilenlineal, Formatierung, eigene Zeichensätze, kurzum eine Menge an Funktionen und Einstellungen sind möglich. 'Suchen und Ersetzen' und eine automatische Trennhilfe runden das Bild über die Textverarbeitung ab. Das Programm läßt einige Voreinstellungen zu, der gleichzeitige Betrieb mit anderen Programmen ist möglich. Die Geschwindigkeit ist nicht besonders hoch, wer jedoch nicht allzu viel Wert auf Schreib- und Darstellungsrekorde legt, findet in KindWords einen idealen Partner, um seine Korrespondenz an den Mann (oder die Frau) zu bringen. Die Rechtschreibhilfe kann bei ausreichendem Speicher komplett ins RAM geladen werden, ist jedoch ebenfalls nicht schnell. Das Wörterbuch kann erweitert werden, Vorschläge sind jedoch schon von Anfang an ausreichend vorhanden. Textformate wie Block- oder Flattersatz, größerer Zeilenabstand, links-, rechtsbündiges oder zentriertes Schreiben läßt sich ohne Umwege über Icons anwählen, dementsprechend leicht ist das Ändern von Texten. Eines noch zum Thema Meckerecke: In unserer Testversion existierte noch ein Zeichensatz namens 'Griechlisch', was anscheinend etwas mit 'Griechlenland' zu tun hat. Falls in neueren Versionen dieser Bug behoben worden ist, hat hier keiner was gesagt, sonst heißt es, den Namen nicht umzuändern, weil sonst das Programm den Zeichensatz nicht mehr erkennt. KindWords besitzt den Status einer zwar etwas behäbigen, aber trotzdem soliden Textverarbeitung, die sich hinter anderen Erzeugnissen dieser Gattung nicht zu verstecken braucht. Noch etwas! KindWords wird nicht nur als Solo-Programm angeboten, sondern auch als Bestandteil des DTP-Programms 'The Publisher', wo die Textverarbeitung im Zusammenspiel mit dem Layout-Programm 'PageSetter' zur Erstellung von Layoutseiten benutzt wird (siehe Test in AMIGA DOS 03/90).

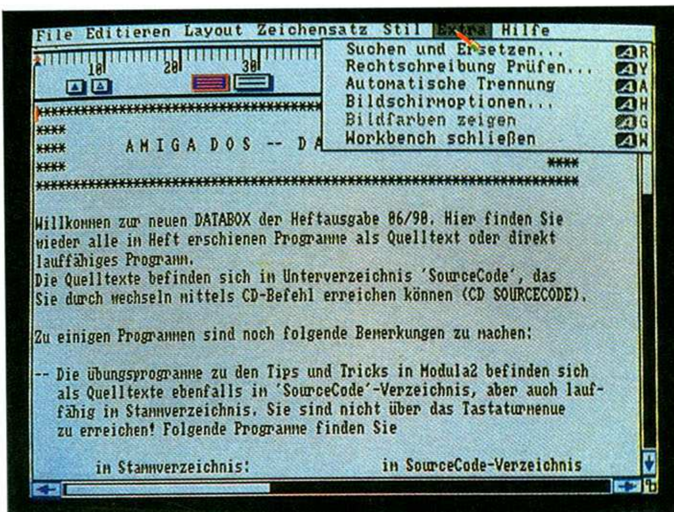


Bild 9. KindWords – die Gemütliche

Excellence

Mit einem Preis, der sich um die 400 DM bewegt, nimmt Excellence einen Platz unter den 'professionellen' Textverarbeitungen ein. Auch was den Leistungsumfang des Produktes anbetrifft, finden sich hier viele Features, die Excellence als Profi-Textverarbeitung ausweisen: Serienbrief-Option, Blockoperationen, verschiedene Schrifttypen, die ferner noch als verschiedene Darstellungsarten (Fett, Kursiv, etc.) zur Verfügung stehen, WYSIWYG-artige Bildschirmdarstellung, automatische Fußnotenverwaltung, Lexikonfunktion, Synonymwörterbuch, Postscript-Ausgabe und einiges mehr. Obwohl an Leistungsdaten an und für sich alles vorhanden ist, was eine gute Textverarbeitung auszeichnet, gibt es Kritikpunkte. So scheint die Verarbeitungsgeschwindigkeit des Programms hinter der Funktionsvielfalt ein Schattendasein zu führen und die Bearbeitung längerer Dokumente gestaltet sich zäh. Excellence benötigt mindestens 1 MByte RAM, läßt sich also auf der Grundversion des Amiga 500 nicht benutzen. Das Handbuch des Produktes ist mit zirka 300 Seiten in deutscher Sprache recht voluminös ausgefallen, hätte unseres Erachtens allerdings besser strukturiert werden können. Excellence ist ein Tausendsassa unter den Textverarbeitungen.

Die Features, die das Produkt beinhaltet, heben es deutlich über den qualitativen Durchschnitt. Die mangelnde Verarbeitungsgeschwindigkeit macht sich jedoch bei längeren Dokumenten sehr störend bemerkbar.

(vb)

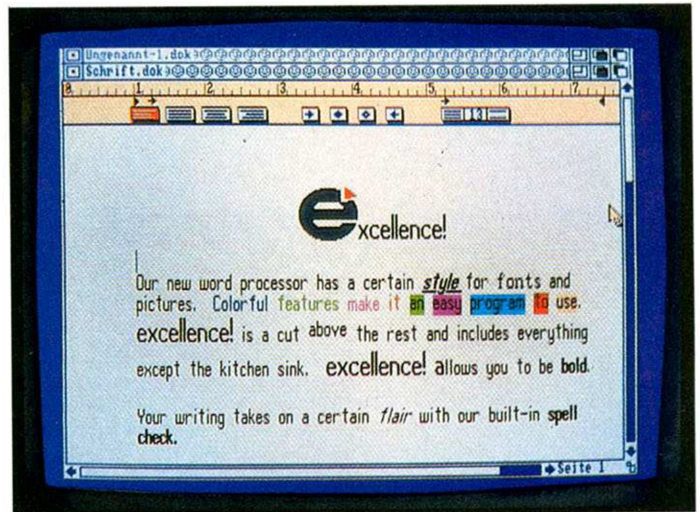


Bild 10. Excellence – die Umfangreiche

AMIGA DOS Info

Name: Wordperfect
Preis: ca. 800 DM
Hersteller: Wordperfect
Mindestspeicher: 1 MByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: gut
- Zeilenlineal: nein
- Blockoperationen: ja + Speichern, Laden, Drucken
- Suchen/Ersetzen: ja + formatierte Texte
- Silbentrennung: ja
- WYSIWYG: eingeschränkt
- Serienbrief: ja
- Lexikon: ja + Vorschläge, suchen
- Fuß-, Endnotenverw.: ja, getrennt vom eigentlichen Text
- Phrasenspeicher: über Makros

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: ja, in regelmäßigen Abständen
- Zusatzfunktionen: Löschen, Rename, Text suchen
- eigene Zeichensätze: 1
- Verwaltung zus. Druckerfonts: ja
- Bilder einlesen: ja
- eigene Druckertreiber: ja, änderbar (insgesamt 258)
- Thesaurus: ja
- mehrspaltiger Text: ja, ein- und ausschaltbar
- Rechnen im Text: ja

AMIGA DOS Info

Name: Beckertext 1.02
Preis: ca. 200 DM
Hersteller: Data Becker
Mindestspeicher: 1 MByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: gut
- Zeilenlineal: nein / Tabulatorzeile
- Blockoperationen: ja + Spalten verschieben
- Suchen/Ersetzen: ja + formatierte Texte
- Silbentrennung: ja
- WYSIWYG: sehr eingeschränkt
- Serienbrief: ja
- Lexikon: ja, sogar im Direkt-(Online-)Betrieb
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: über Funktionstasten

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: ja, der zuletzt vorhandene Text
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: 2
- Verwaltung zus. Druckerfonts: ja
- Bilder einlesen: ja, eingeschränkte Wiedergabe
- eigene Druckertreiber: ja, änderbar
- Thesaurus: nein
- mehrspaltiger Text: ja, nur beim Ausdruck
- Rechnen im Text: ja

AMIGA DOS Info

Name: Excellence
Preis: ca. 400 DM
Hersteller: Micro-Systems
Mindestspeicher: 1 MByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: befriedigend
- Zeilenlineal: ja
- Blockoperationen: ja
- Suchen/Ersetzen: ja
- Silbentrennung: ja
- WYSIWYG: ja
- Serienbrief: ja
- Lexikon: ja + Vorschläge
- Fuß-, Endnotenverw.: ja
- Phrasenspeicher: nein

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: ja, der zuletzt vorhandene Text
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: ja
- Verwaltung zus. Druckerfonts: nein
- Bilder einlesen: ja, eingeschränkte Wiedergabe
- eigene Druckertreiber: ja, änderbar
- Thesaurus: ja
- mehrspaltiger Text: ja
- Rechnen im Text: ja

AMIGA DOS Info

Name: Textomat Amiga
Preis: ca. 100 DM
Hersteller: Data Becker
Mindestspeicher: 512 kByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: befriedigend
- Zeilenlineal: nein / Tabulatorzeile
- Blockoperationen: ja + Spalten verschieben
- Suchen/Ersetzen: ja + formatierte Texte
- Silbentrennung: ja
- WYSIWYG: sehr eingeschränkt
- Serienbrief: nein
- Lexikon: nein
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: über Funktionstasten

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: ja, der zuletzt vorhandene Text
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: 2
- Verwaltung zus. Druckerfonts: nein
- Bilder einlesen: ja, eingeschränkte Wiedergabe
- eigene Druckertreiber: ja, änderbar
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein

AMIGA DOS Info

Name: Vizawrite 2.0
Preis: ca. 200 DM
Hersteller: Viza Software
Mindestspeicher: 512 kByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: befriedigend
- Zeilenlineal: ja
- Blockoperationen: ja
- Suchen/Ersetzen: ja
- Silbentrennung: ja
- WYSIWYG: ja
- Serienbrief: ja
- Lexikon: nein
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: über Textbausteine

Besonderes:
- Voreinstellungen: ja, viele
- Sicherheitskopie: nein
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: nein
- Verwaltung zus. Druckerfonts: ja
- Bilder einlesen: nein
- eigene Druckertreiber: ja, änderbar
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein



Marktübersicht Textverarbeitungen

AMIGA-DOS-Übersicht

Wir haben hier Informationen über bekannte Textverarbeitungen in Kurzform zusammengefaßt, damit Sie sich selbst ein Bild machen können, was diese oder jene Textverarbeitung zu leisten vermag. Zu einigen Angaben hier noch einige Bemerkungen:

Bei den Blockoperationen sind in dem bewertenden "Ja" folgende Funktionen eingeschlossen: Kopieren, Versetzen, Löschen, Ausschneiden.

Beim Lexikon gelten folgende Funktionen mit der Bewertung "Ja" als enthalten: Korrigieren, Wort aufnehmen, Überlesen.

Die Zusatzfunktionen enthalten nicht die üblichen wie Laden und Speichern.

AMIGA DOS Info

Name: UBM-Text 2.3
Preis: ca. 150 DM
Hersteller: UBM-Drecker
Mindestspeicher: 512 kByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: befriedigend
- Zeilenlineal: ja
- Blockoperationen: ja
- Drucken, Speichern, + Laden
- Suchen/Ersetzen: ja
- Silbentrennung: ja
- WYSIWYG: sehr eingeschränkt
- Serienbrief: ja
- Lexikon: nein
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: über Funktionstasten

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: nein
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: 2
- Verwaltung zus. Druckerfonts: über Druckercodes
- Bilder einlesen: ja, über externes Programm
- eigene Druckertreiber: ja, änderbar
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein

AMIGA DOS Info

Name: Transcript
Preis: ca. 100 DM
Hersteller: Gold Disk
Mindestspeicher: 512 kByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: englisch
- Geschwindigkeit: befriedigend
- Zeilenlineal: nein
- Blockoperationen: ja
- Suchen/Ersetzen: ja
- Silbentrennung: ja (englisch)
- WYSIWYG: ja
- Serienbrief: nein
- Lexikon: ja, über eigenes Programm
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: durch Makros

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: nein
- Zusatzfunktionen: nein
- eigene Zeichensätze: 1
- Verwaltung zus. Druckerfonts: nein
- Bilder einlesen: nein
- eigene Druckertreiber: nein
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein

AMIGA DOS Info

Name: Prowrite 2.01
Preis: ca. 200 DM
Hersteller: new Horizons
Mindestspeicher: 1 MByte
Kopierschutz: nein
Bedienung: Maus, Tastenkombinationen

Leistungen:
- Handbuch: englisch
- Geschwindigkeit: befriedigend
- Zeilenlineal: ja
- Blockoperationen: ja
- Suchen/Ersetzen: ja
- Silbentrennung: nein
- WYSIWYG: ja
- Serienbrief: ja
- Lexikon: ja + Vorschläge, Blättern
- Fuß-, Endnotenverw.:nein
- Phrasenspeicher: nein

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: nein
- Zusatzfunktionen: Löschen
- eigene Zeichensätze: nein
- Verwaltung zus. Druckerfonts: nein
- Bilder einlesen: ja, mit Textumbruch
- eigene Druckertreiber: nein
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein

AMIGA DOS Info

Name: Kindwords
Preis: ca. 150 DM
Hersteller: Disk Company
Mindestspeicher: 1 MByte
Kopierschutz: nein
Bedienung: Maus, Funktionstasten, Tastenkombinationen

Leistungen:
- Handbuch: deutsch
- Geschwindigkeit: befriedigend
- Zeilenlineal: nein
- Blockoperationen: ja
- Suchen/Ersetzen: ja
- Silbentrennung: ja
- WYSIWYG: ja
- Serienbrief: ja
- Lexikon: ja + Vorschläge
- Fuß-, Endnotenverw.: nein
- Phrasenspeicher: nein

Besonderes:
- Voreinstellungen: ja, wenige
- Sicherheitskopie: nein
- Zusatzfunktionen: nein
- eigene Zeichensätze: 2
- Verwaltung zus. Druckerfonts: nein
- Bilder einlesen: nein
- eigene Druckertreiber: ja, nicht änderbar
- Thesaurus: nein
- mehrspaltiger Text: nein
- Rechnen im Text: nein

Die am meisten verbreiteten Anwendungen in der Computerbranche sind wahrscheinlich die Dateiverwaltungssysteme. Ein Programm zum Verwalten und Manipulieren von Daten auf dem Amiga liegt mit Datamat vor. Das Buch erklärt die beiden Versionen Datamat Plus und Datamat Professional und gibt hilfreiche Tips und Tricks für den Umgang damit.

Datamat Know How

Das Buch gliedert sich in vier Kapitel, wobei das erste dem Neueinsteiger in das Programm Datamat gewidmet ist. So werden vorerst Grundsätzlichkeiten wie die Installation des Programms, Tastenbelegung und Voreinstellungen geklärt.

Das zweite Kapitel befaßt sich mit den Grundlagen der Datei-Erstellung. Anhand von Screenshots und vielen Erklärungen ist es dem Leser ein leichtes, das Beschriebene zu verstehen und nachzuvollziehen. Es wird über Pulldown-Menüs eine einfache Bildschirmmaske erstellt, wobei der Umgang mit verschiedenen Datensätzen nicht vergessen wird.

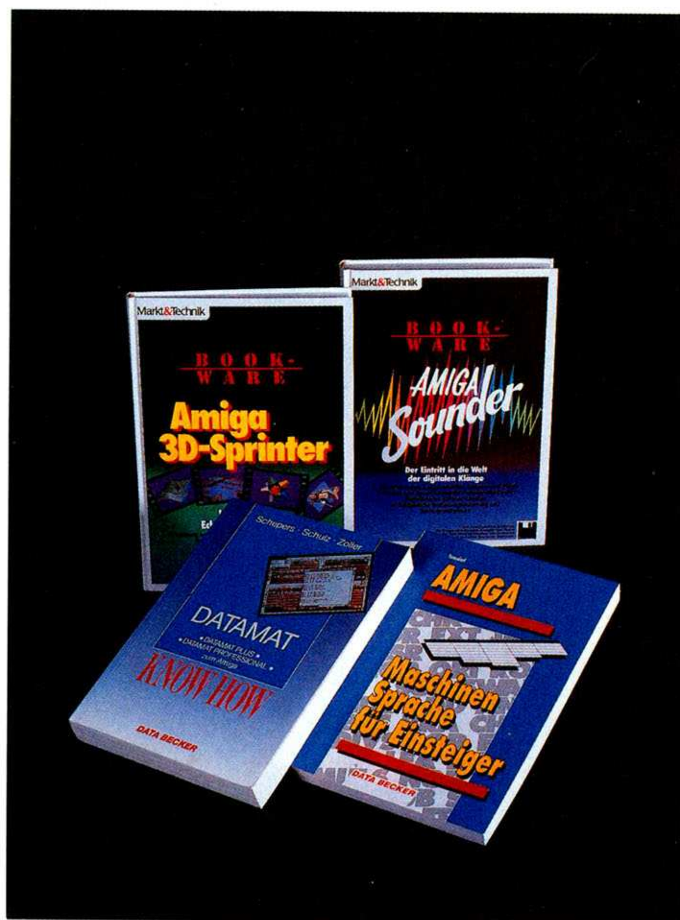
Im dritten Teil des Buches wird auf die Erstellung von Bildschirmmasken mittels Maskeneditoren eingegangen. Das Buch zeigt anschließend, wie man Standardmasken für die Bildschirm- und Druckerausgabe ändert. Ähnlich dem im MS-DOS etablierten dBase ist auch in Datamat Professional eine Programmiersprache implementiert. So führt das Buch einen Exkurs in diese Programmiersprache namens Profil. Vorab wird dem Benutzer jedoch der Umgang mit dem dafür vorgesehenen Editor plausibel gemacht. Das Buch geht bei der Erklärung der Programmiersprache sehr geschickt vor. So werden als erstes die Variablendeklarationen definiert, um dann eine Datei laden und einen Index anlegen zu können. Einzelne Unterprogramme verstärken die Übersicht.

Die Unterprogramme werden dann in das Hauptprogramm eingebunden, so daß der Leser, wenn er das Kapitel durchgearbeitet hat, keine Probleme mit der Programmiersprache Profil haben sollte.

In dem mehr als einhundert Seiten umfassenden Anhang wird die gesamte Befehlsliste der Programmiersprache Profil lexikalisch aufgeführt. Die Befehle sind zuerst mit der entsprechenden Syntax, dann einer Erklärung und schließlich einem kleinen Beispiel versehen. Zusätzlich findet man hier die so mühsam erarbeiteten Einzelprogramme in einem Listing vereinigt. Aufgrund vieler Listingdokumentationen wird der Programmierfluß noch einmal vertieft. Das Stichwortverzeichnis schließt letztendlich ein gelungenes Buch zum Umgang mit Datamat ab.

(Jürgen Seibel/br)

Name: Datamat Know How
Autoren: Schepers, Schulz, Zoller
Verlag: DATA-Becker
ISBN: 3-89011-339-7
Preis: 39,– DM



Das erste Hardwarebuch, an dem auch Bastelfreaks ihre wahre Freude haben werden, liegt mit dem Werk Amiga-Sounder von Markt & Technik vor. Wer Spaß am Samplen hat, kommt mit diesem Buch voll auf seine Kosten.

Amiga-Sounder

Zuerst wird man mit der auf zwei Disketten mitgelieferten Software auf leicht verständliche Weise vertraut gemacht. Da ist zunächst das Hauptprogramm Amiga-Sounder, das in seiner Art durchaus mit dem 'Audiomaster2' vergleichbar ist und diesem Vergleich ohne weiteres auch standhält. Einziges Manko sind die stellenweise sehr langen Wartezeiten, die sich jedoch immer lohnen. Vor allen Dingen besticht der schnelle File-Requester, der problemlos mit einer Harddisk zusammenarbeitet.

Das Programm 'Sampler' erlaubt im Gegensatz zum Amiga-Sounder das Digitalisieren von maximal vier Kanälen gleichzeitig(!). Leider ist es nicht möglich, beispielsweise ein in Stereo digitalisiertes Musikstück so abzuspeichern, daß es mit 'Audiomaster2' geladen und bearbeitet werden könnte. Mit der uns vorliegenden Software des AMAS-Samplers ist dies jedoch möglich.

Der große Vorteil des 'Samplers' ist sein geringer Speicherbedarf. Somit steht der gesamte freie Arbeitsspeicher für digitale Sounds zur Verfügung. Es war ohne weiteres möglich, sage und schreibe viereinhalb Minuten Musik in annehmbarer Qualität in drei MByte Speicher unterzubringen. Solche Stücke lassen sich mit dem Programm 'Player' vom CLI oder von der Workbench aus abspielen (leider nur Mono).

Die Sampling-Rate sollte in diesem Fall allerdings nicht allzuhoch gewählt werden, da der 'Player' immer wieder von Diskette nachlädt, während er das Stück abspielt. Eine Harddisk ist dagegen jeder Geschwindigkeit gewachsen. Insgesamt benötigt dieses Programm nur zirka 100 kByte Speicher und ist deshalb auch für einen Amiga 500 mit 512 kByte Arbeitsspeicher geeignet. Des weiteren stehen ein Programm zur direkten Ausgabe der momentan eingelesenen Daten und ein Echtzeit-Echo (oder auch Hall) zur Verfügung.

Die Hardware ist relativ einfach gehalten: Zur Bestückung der mitgelieferten Platine sind nur zwölf Bauteile notwendig. Die Kosten belaufen sich auf rund 70 DM, da bereits der A/D-Wandler von Analog-Devices etwas über 40 DM kostet. Die mitgelieferte Platine ist so klein gehalten, daß sie direkt an den Parallel-Port gesteckt werden kann. Leider ist der Aufbau der Platine etwas problematisch, da die Lötpunkte ebenfalls sehr klein sind und leicht abreißen. Vor allem diejenigen, die im Löten nicht viel Erfahrung haben, sollten sich vorsehen. Durch den Platzmangel ist es außerdem schwer, die Kondensatoren, die zur kapazitiven Kopplung der NF-Signale benötigt werden, unterzubringen, da man so schmale Ausführungen nur sehr selten bekommt. Hinsichtlich dieser Kondensatoren und einiger anderer Kleinigkeiten ist der Bestückungsplan sehr unübersichtlich. Mit Hilfe der vier Fotografien des Samplers kann man sich aber den Rest "zusammenreimen".

Beim Aufbau sind eigentlich keine schwerwiegenden Feh-

ler zu machen. Wenn man diese Arbeit erfolgreich beendet hat, geht es ans Abgleichen. Hierfür ist ebenfalls ein Programm auf den beiliegenden Disketten vorhanden. Auf diese Weise ist der Abgleich relativ einfach, vorausgesetzt, man hat Geduld und regt sich nicht auf, wenn sich der soeben eingestellte Kanal wieder um einiges verstellt, während man die anderen drei einjustiert. Fingerspitzengefühl ist hier gefragt.
Im zweiten Teil des Buches erklärt der Autor dann ausführlich die wichtigsten Routinen der vorgestellten Programme. Der dritte Teil beschäftigt sich mit dem nötigen Hintergrundwissen. Dabei geht der Stoff gut ans Eingemachte, denn mit Integralen wird nicht gespart – pure Theorie vom Feinsten.
Abschließend bleibt noch die schönste aller Erfahrungen mit dem Amiga-Sounder zu erwähnen: Er funktioniert ebensogut mit dem 'Audiomaster2' (nur Kanal drei), wie mit der Software des AMAS-Samplers (nur Kanal eins)!
Wer für insgesamt 170,– DM ein hervorragendes Buch und einen noch besseren Sampler erwerben möchte und vor ein bißchen Bastelei nicht zurückschreckt, dem kann man dieses Buch wärmstens empfehlen.

(Bernd Rudolf/br)

Name: Amiga-Sounder
Autor: Heiko Knappe
Verlag: Markt & Technik
ISBN: 3-89090-709-1
Preis: 98,– DM

Ein interaktives Echtzeit-Animationspaket, welches aus einem 150 Seiten umfassenden Buch und zwei Disketten besteht, liegt mit dem Werk Amiga 3D-Sprinter vor. Diese sogenannte 'Book-Ware' ist nach Definition des Verlages Markt & Technik "Profi-Software zum Buchpreis". Hält 3D-Sprinter was es verspricht?

Amiga 3D-Sprinter

Das Programm 3D-Sprinter erlaubt unter anderem die einfache Erstellung von Objekten. Würfel, Kreise, Drehkörper und Funktionsgraphen sind nur einige von den vielen Objekten, die beliebig zusammengesetzt werden können. Diese Objekte erscheinen dem Betrachter auf dem Bildschirm als Drahtgittergrafik oder als Objekte mit ausgefüllten Flächen und Schatten. Licht und Kameraposition können in einem Animationskript festgelegt werden und ergeben beispielsweise durch Kamerafahrten in einer 3D-Welt verblüffende Effekte.
Die deutsche Programmversion unterstützt alle Grafikauflösungen des Amiga, wobei bis zu 64 Farben gleichzeitig dargestellt werden können. Was kann das Grafikpaket noch? Nun, auf die Darstellung der generierten Objekte als Drahtgittermodell, als Plotterzeichnung, schattiert mit und ohne Schlagschatten sind wir ja schon eingegangen. Die Objekte können dann über die Tastatur, per Maus oder mit Hilfe von editierbaren Skript-Dateien manipuliert werden. Das Drehen, Bewegen, Vergrößern und Verkleinern ganzer Szenen erfolgt direkt am Bildschirm. Die Animationen und Spiegelungen werden, je nach Rechnerkonfiguration, in Echtzeit berechnet, was natürlich einiges an Zeit beanspruchen kann, wenn man nur über einen Rechner in der Grundversion (68000-Prozessor) verfügt.
Das beiliegende Buch beinhaltet einen kompletten Einführungskurs in Computer und Programm. Der Computerneuling wird buchstäblich vom Autor an die Hand genommen und erlernt an Hand von kleineren Beispielen (vom Einschalten bis zum Schreiben von Animationsskripten) alles, was er über die Anwendung von Amiga 3D-Sprinter wissen muß.
Das dritte Kapitel befaßt sich mit der Theorie der 3D-Grafik. In diesem wichtigen Abschnitt versteht es der Autor geschickt den zugegeben etwas trockenen, aber sehr wichtigen Stoff an den Mann (beziehungsweise die Frau) zu bringen.
Dennoch hat auch dieses Paket seine Schwächen. Als wir komplett berechnete und abgespeicherte Animationen ansehen wollten, fiel natürlich sofort die Geschwindigkeit der Animationen auf. Das Abspielprogramm ist so träge, daß man es kaum wagen kann, einen Körper mit gefüllten Flächen zu animieren. Hier gilt es, in schon jetzt angekündigten zukünftigen Versionen von 3D-Sprinter Verbesserungen vorzunehmen.
Das Programm läßt sich ohne weiteres auf einer Festplatte installieren und unterstützt die Ausgabe auf einen Plotter. Zum Schluß muß ich noch auf die (glücklicherweise) minimale Hardware-Anforderung eingehen. Grafikprogramme benötigen meist einen möglichst großen Arbeitsspeicher, so auch 3D-Sprinter. Ein MByte und Kickstart 1.2 (oder höher) sollten es für den reibungslosen Ablauf also schon sein, ein eingeschränkter Betrieb ist jedoch auf auf einem 512-kByte-Amiga möglich.

(Ingmar Reyer/br)

Name: Amiga 3D-Sprinter
Autoren: Glaeser/Grohser
Verlag: Markt&Technik
ISBN: 3-89090-109-3
Preis: 98,– DM

Wohin man auch hört, überall ist das Wort Maschinensprache oder der Begriff Assembler im Umlauf – jedenfalls dort, wo man über Leistungen von Computern redet. Denn anscheinend ist wirklich nur die Maschinensprache es wert, auf Computern wie dem Amiga eingesetzt zu werden, wenn man den 'Freaks' und 'Kennern' glauben darf. Als Einsteiger scheut man automatisch vor dem Wort 'Maschinensprache' zurück, weil man zu wenig von der Materie kennt und jeden Eingriff in das System des Rechners scheut. Um das heikle Thema in Einsteigerkreisen an den Mann (respektive die Frau) zu bringen, hat es sich Autor Manfred Tornsdorf zur Aufgabe gemacht, mit wenig Aufwand viele Kenntnisse zu übermitteln.

Maschinensprache für Einsteiger

Und so beginnt das Buch quasi bei Adam und Eva – oder besser gesagt bei 'MOVE' und 'ADD'. Anhand von kleinen Beispielen, die an den Seka-Assembler der Firma Kuma angepaßt sind, werden die ersten Maschinensprache-Befehle erlernt, jedoch nicht ohne die obligatorische Einführung in besagten Assembler. Die Beispielprogramme sind so gehalten, daß sie auch auf anderen Assemblern schnell eingesetzt werden können. Leider wurde der Devpac von Hisoft bei der Vorstellung anderer Assembler nicht berücksichtigt, obwohl auch er eine weite Verbreitung gefunden hat und auch in überarbeiteter Version verkauft wird. Aber zurück zum Buch.
Die ersten Programme zeigen schnell den Umgang mit Additionen und Subtraktionen, sowie den Einsatz von bedingten Verzweigungen und Sprüngen. Auch das Austesten einzelner Bits wird dem Leser kundgetan, nach kurzer Zeit ist er in der Lage, Programme mittels Mausklick zu beenden oder die Power-LED durch Programmierung aus- und einzuschalten.
Der Vorteil des Buches liegt darin, den Leser nicht sofort mit Amiga-spezifischen Daten zu erschlagen, Window- und Intuition-Programmierung schließen sich erst an, wenn der Leser schon einigermaßen sicher auf dem rutschigen Parkett namens Maschinencode stehen kann. Und selbst dann wird Wert darauf gelegt, erstens nicht zuviel Verwirrendes und Unnötiges in die Beispiele zu packen und zweitens den Leser zum Mitdenken aufzufordern, indem der Programmablauf Schritt für Schritt erklärt wird. Auch die Programmierung unter Zuhilfenahme des Betriebssystems wird angegangen, wobei außer Einlesen und Ausgeben von Texten in Windows auch Diskettenoperationen behandelt werden.
'Maschinensprache für Einsteiger' ist kein Buch für die, die sich zwar als Anfänger fühlen, aber schon Erfahrung mit Maschinensprache haben, sondern eher für diejenigen, denen Assembler und Maschinensprache auf dem Amiga bisher böhmische Dörfer waren. Zwar ist das Buch manchmal etwas oberflächlich und behandelt ein Thema nicht so ausführlich, wie es sein sollte, jedoch ist es ein Buch, das neben dem Computer liegen sollte, wenn es zur Sache geht.

(jb/br)

Name: Maschinensprache für Einsteiger
Autor: Manfred Tornsdorf
Verlag: DATA Becker
ISBN: 3-89011-172-6
Preis: 39,– DM

AMIGA DOS

Endlich keine Listings mehr abtippen!

Nicht bei allen Programmen ist es mit drei Zeilen getan – gute Routinen und praktische Funktionen brauchen ihren Platz. Und bisweilen lassen sich auch lange Datenblöcke nicht vermeiden, ganz zu schweigen von Hexdumps und Assemblerlistings.
Schonen Sie Ihre Augen und schlagen Sie sich nicht die Nacht mit Abtippen um die Ohren. – Auf der Databox zum Amiga DOS-Heft finden Sie alle Listings als ASCII-File: passend für jeden Texteditor, den Amiga-BASIC-Interpreter, Makro-Assembler oder einen Compiler für C und Modula-2.

Alle Programme sofort nutzen

Da ist er nun endlich – der Trick oder das Programm, auf das Sie schon so lange gewartet haben! Zu allem Unglück ist das Listing aber in Modula-2 oder C, jedenfalls in einer Sprache, zu der Sie keinen Compiler haben, um ein lauffähiges Programm herzustellen.

Auch in diesem Fall hilft Ihnen die Databox von Amiga DOS aus der Patsche: Neben den Quelltexten im ASCII-Format finden Sie jeweils auch das fertige, lauffähige Programm. Sie brauchen es nur von der Databox-Diskette aus zu starten.

DATA

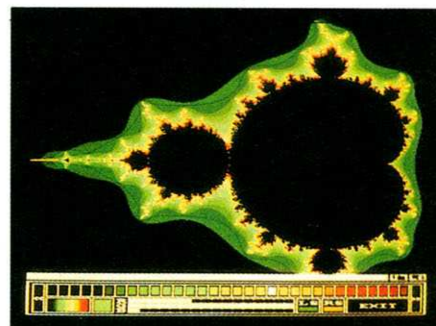
Keinen Ärger mehr mit Tippfehlern

Wer kennt das nicht, wenn das Programm nach dem Eintippen nicht läuft oder der Rechner gar abstürzt. Besonders gemein sind auch Fehler, die erst nach Wochen bei einer bisher nicht gebrauchten Funktion zu Tage treten, oder wenn der Druckfehlerteufel am Werke war.

Zermartern Sie sich nicht den Kopf, bis Sie die falsche Zahl im Datafeld gefunden haben. – Alle Dateien auf der Databox zur Amiga DOS sind vom Autor und der Redaktion auf Fehlerfreiheit geprüft und im dazugehörigen System "probegelaufen".



External Boot – booten wir doch einfach von Laufwerk »df2:«



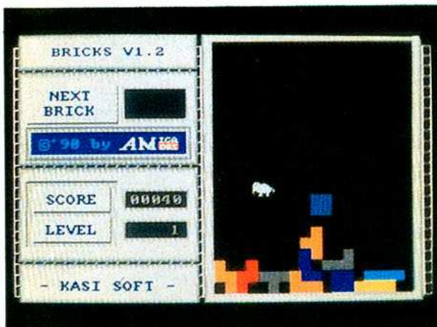
RGB-Change – ändern Sie die Farben Ihrer Bilder

Alle Listings und Programme auf Diskette – Computer einschalten – Diskette einlegen – los geht's

2 Disketten voller Programme

BOX

Basic	C	Modula-2	Assembler	Lauffähiges Programm	Inhalt der Databox AMIGA DOS 7/90
X				X	Tetris – das erste Computerspiel aus Rußland – gab den Anreiz zu BRICKS. Erproben Sie Ihre Künste
X				X	RGB-Change – mit diesem Programm können Sie die Farben eines IFF-Bildes ändern
	X			X	Das Sprite-Editor-Projekt Mit dem Farb-Requester fällt die Auswahl leicht
	X			X	Ein Demoprogramm macht den Umgang mit der ARP-Library etwas verständlicher
		X			Tips und Tricks zu Prozessen und ausführbaren Programmen
		X		X	Ein Drucker-Utility hilft bei der Arbeit
		X		X	Mini-Datenverwaltung als Einstieg in den Modula-Kurs
			X	X	Der Vektorverbieger hilft Ihnen bei der Anpassung alter Programme
			X	X	Im Blitter-Kurs geht es diesmal um die Vektorgrafik
			X	X	External Boot – damit wird das Booten von externen Laufwerken zum Kinderspiel
		X		X	**Und zusätzlich auf der Diskette:** WordMate – ein Vokabeltrainer, der lernen leicht macht



Bricks – schnell denken und lenken

24,– DM
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		**Ausland:**	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Zahlungsweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen in das Ausland ist Nachnahme nicht möglich.)

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Die PD-Werkzeugkiste

Nach dem Kauf eines Computers ist meist totale Ebbe im Geldbeutel. Was tun? Ohne Software nutzt auch die beste Hardware nichts. Wer kennt die Situation nicht? Man fragt kurzerhand einen Bekannten und kopiert illegal teure Programme. Meist fehlt trotzdem noch eine auskunftgebende Anleitung, und schon ist der große Frust da. Es geht aber auch anders...

Gute Software muß nicht unerschwinglich sein, das beweisen schon seit geraumer Zeit diverse Public-Domain- und Shareware-Bibliotheken. Leider werden Begriffe wie Public Domain und Shareware oft falsch verwendet und gleichgesetzt. Wir wollen kurz versuchen ein wenig Licht ins Dunkel zu bringen.

Public Domain

Public-Domain-Programme sind Softwareprodukte, bei denen der Autor seine Programme unter Verzicht auf Copyright und Vermarktung kostenlos einer breiten Öffentlichkeit zur Verfügung stellt. Diese Programme dürfen ohne weitere Genehmigung kopiert, getauscht, genutzt und auch verändert werden.

Shareware

Unter den Begriff Shareware fallen alle Programme, die der Anwender vor dem Kauf oder der Registrierung ausgiebig und kostenlos auf Herz und Nieren testen darf. Nach amerikanischem Recht, denn in den USA ist diese Idee entstanden, verbleiben alle Rechte inklusive der Vermarktung und des Copyrights beim Entwickler beziehungsweise beim Anbieter. Die Programme dürfen vom Anwender nicht verändert, dem jeweiligen Disketteninhalt nichts entnommen oder hinzugefügt werden. Die Vervielfältigung und Weitergabe der Diskette ist sogar erwünscht, um möglichst viele Anwender mit dem Produkt bekannt zu machen.

Bei der Weitergabe dürfen nur geringe Beträge für die Diskette und den Kopieraufwand in Rechnung gestellt werden, ein Verkauf der Shareware ist verboten. Sagt nun ein Programm dem Anwender zu und er möchte es nutzen, so läßt er sich beim Entwickler oder Anbieter registrieren. Dafür zahlt er eine sogenannte Shareware-Gebühr und erhält als Gegenleistung die jeweils neueste Version des Programms, eine ausführliche Anleitung, die Möglichkeit eines späteren Updates und Unterstützung durch den Entwickler bei Problemen im Zusammenhang mit dem Produkt.

Leider trifft man viel zu oft bei verschiedenen PD-Serien auf eine Mischung diverser Public-Domain- und Shareware-Programme, so daß eine Klassifizierung der jeweiligen Diskette nicht gelingt. Hier greift die Shareware-Reihe mit Namen Share ein, die wir in der letzten Ausgabe schon vorgestellt haben. Auf den jeweiligen Disketten befinden sich reine Shareware-Programme, wobei ein Teil des Verkaufspreises den Autoren zugute kommt.

Doch genug der Vorrede: Beginnen wollen wir unsere Exkursion in das Reich der Public Domain und Shareware diesmal mit der Share 5.

Diskey, das Multitalent

Diskey ist ein in der Programmiersprache C entwickelter Diskettenmonitor von Angela Schmidt, der mit vielen Features überzeugen kann. Er arbeitet block- bzw. sektorweise, ist also kein File-Monitor, bei dem man einen Dateinamen eingeben muß und das entsprechende Programm dann editiert.

Die verschiedenen Unterfunktionen lassen sich durch eine in deutscher Sprache gehaltene und benutzerfreundlich aufgebaute Oberfläche mittels Pulldown-Menüs beziehungsweise durch leicht einprägsame Tastenkombinationen aufrufen.

Das Programm überzeugt vor allem durch neue Einfälle, so daß es bisherige Diskettenmonitore leicht in den Schatten stellt. Man findet beispielsweise eine Routine, die gewünschte Tracks formatiert, ohne deren Inhalt zu löschen. Dabei wird der Disketteninhalt kurzfristig in das RAM kopiert und nach dem Formatieren wieder auf der Diskette gespeichert. Damit entfällt das lästige Anlegen einer Sicherheitskopie bei der Behebung von Disk-Errors.

Für diejenigen, die keine oder nur veraltete Virenschutzprogramme besitzen, stellt Diskey zudem eine nützliche Ergänzung dar. Neu eingelegte Disks werden gleich auf eine Vielzahl von Viren (Lamer-Exterminatoren eingeschlossen) untersucht. Sollte sich ein Virus auf der Diskette befinden, wird eine entsprechende Meldung auf den Bildschirm gebracht. Allerdings ist diese Funktion nur eine Sekundärfunktion. Ein hundertprozentiger Verlaß, daß alle Viren erkannt werden, ist somit nicht gewährleistet.

Alle Funktionen, die einen guten Diskettenmonitor auszeichnen, sind in Diskey integriert, so beispielsweise Such-, Lese-, Schreib-, Checksummen-, Formatierungs-, Füll-, Follow- und Initialisierungsfunktionen.

Bei der Option 'INIT DISK' wird eine Diskette im Schnellverfahren formatiert, das heißt, die Daten sind zwar noch auf der Diskette enthalten, allerdings sind die Zeiger im Root- und Boot-Block beziehungsweise in der BAM auf Null gesetzt, so daß der Computer die Diskette als frisch formatiert betrachtet. Deshalb wurde für diese und einige andere Funktionen, so auch bei FORMAT und SCHREIBEN, eine spezielle Sicherung eingebaut, um Datenverluste zu vermeiden. Die einzelnen Tracks, Sektoren und Blocks lassen sich alle einzeln über Gadgets ansteuern und mittels den Tasten ' + ' und '-' “vor-“ beziehungsweise “zurückblättern“. Durch zwei Gadgets können der Root- und Boot-Block direkt angewählt werden. Die Blocks lassen sich über ein Fenster editieren, in dem der momentane Block in Longwords dargestellt wird. In zwei anderen Windows erscheint der Block als ASCII-Code. Der Benutzer hat nun die Möglichkeit, zwischen ASCII- und Hex-Editierung zu wählen. Über die Diskey-Preferences lassen sich eine Vielzahl von Parametern, so zum Beispiel auch die Sicherung der Befehle FORMAT/INIT DISK oder die Farben von Diskey individuell einstellen.

Außerdem sollte noch auf die sehr ausführliche und verständlich gehaltene Anleitung hingewiesen werden. Wer also auf einen professionellen Diskettenmonitor nicht verzichten möchte, der bekommt mit Diskey ein Produkt aus der Shareware-Reihe, das kommerziellen Produkten durchaus Contra zu geben weiß. Und nicht zuletzt dadurch wird das Statement widerlegt, Frauen sollten von Computern die Finger lassen, denn Angela Schmidt hat mit Diskey einen echten Treffer gelandet.

(Timo Siebert/br)



Diskey ist ein mit vielen Features überzeugender Diskettenmonitor, der es leicht mit professionellen Produkten aufnehmen kann

Name: Share 5
Quelle: Herrmanns & Kommelter

So mancher Computerbesitzer oder -benutzer begnügt sich mit den gegebenen Zeichensätzen seines Druckers nach einiger Zeit bestimmt nicht mehr. Neue Fonts müssen her, doch woher nehmen, wenn nicht stehlen? Am besten wäre es sogar, wenn man selbst Zeichensätze, den eigenen Bedürfnissen entsprechend, entwerfen könnte. Abhilfe können hier verschiedene Fonteditoren schaffen, doch meist ist das Problem in der Anpassung an den angeschlossenen Drucker.

Schriftsätze selbst gestalten

Auf der Share 6 befindet sich ein Programm namens **Nase,** das in erster Linie auf den NEC P6 zugeschnitten ist, aber auch die Kompatiblen machen kaum Schwierigkeiten. Mit Hilfe dieses Programms ist der Benutzer in der glücklichen Lage, einen Schriftsatz selbst zu gestalten und mit dem Drucker zu nutzen. Dies geschieht, indem die vom Benutzer definierten Zeichensätze in den Puffer des Druckers geladen werden, so daß der Drucker auf den neuen Schriftsatz zurückgreifen kann.
Nase wird über ein Pulldown-Menü mit drei Optionen gesteuert. Der Menüpunkt "Zeichensatz" befaßt sich, wie der Name schon sagt, ausschließlich mit Zeichensätzen. Hier können einzelne Fonts geladen und gesichert werden. Umfangreich ist in dem Menü der Unterpunkt "Editieren". Nach der Wahl der Schriftart taucht ein Eingabefeld auf, in das Sie per ASCII-Code den zu modifizierenden Buchstaben eintragen müssen. Danach breitet sich ein Gitterfeld aus, das den Buchstaben in Pixel gliedert, so daß Sie die Umgestaltung per Mausklick durchführen können. Unter dem Gitterfeld befindet sich der Buchstabe in Normalgröße. Falls Ihnen der neu erstellte Buchstabe zusagt, wird er in die "neue" ASCII-Tabelle mit dem Gadget "Buchstabe übernehmen" eingebunden. "Konvertieren" erlaubt die Umwandlung von normalen Schriftsätzen zum Beispiel in Draft, -LQ10 oder gar in Schönschriften. Hierzu öffnet sich ein File-Requester, aus dem Sie nun den zu bearbeitenden Font auszuwählen haben. Das zweite Menü "Drucker" installiert den neu definierten Zeichensatz im Drucker. Zusätzlich läßt sich der soeben erstellte Zeichensatz dann ausdrucken.
Das Menü "Zeichen" bietet nichts Aufregendes, bis auf die Tatsache, daß sich die einzelnen Buchstaben in alle Himmelsrichtungen verschieben lassen.

(Jürgen Seibel/br)

Name: Share Nr.6
Quelle: Herrmanns & Kommelter



Scratcher ist ein Programm zum Erstellen von Musikstücken unter Zuhilfenahme von Samples. Ein Demo dieses Programms befindet sich auf der A.U.S.T.R.I.A 21

Auf der A.U.S.-T.R.I.A. 21 befindet sich ein Demo des Shareware-Musikprogramms Scratcher, einem interaktiven Sampleplayer, der eine totale Kontrolle über die Art und Weise, wie ein Sample abgespielt wird, geben soll.
Man kann mit diesem Programm einen digitalisierten Klang beispielsweise als Rhythmus, als Fill-In, als Instrument oder aber auch nur als normales One-Shot-Sample verwenden. Abhängig von der ausgewählten Art gibt es verschiedene Kontrollmöglichkeiten wie Start und Stop, Setzen und Spielen von einer Markierung, vorwärtiges und rückwärtiges Abspielen von Samples und vieles andere mehr. Bis zu 30 verschiedene Samples können gleichzeitig in den Speicher geladen und abgespielt werden.

Scratcher – Supersound durch Samples

Der Scratcher ist jedoch kein Sample-Editor, was bedeutet, daß geladene Samples nicht verändert werden können. Wie bereits erwähnt, verwendet das Programm die Samples in vier verschiedenen Arten: als normales Sample, als Instrument, als Rhythmus oder als Fill-In. Verwendet man ein Sample als Instrument, so kann man die Amiga-Tastatur als Keyboard benutzen und so mit dem geladenen digitalisierten Sound Melodien erzeugen.
Im Rhythmus-Modus wiederholt sich der Sound immer wieder, bis er gestoppt wird. Es läßt sich somit eine Hintergrundmusik erzeugen, zu der man beliebige Samples mischen kann.
Fill-Ins sind eine Variante des Rhythmus, wie man sie von einigen Schlagzeugsoli her kennt. Nach dem Abspielen eines Fill-Ins kehrt der Scratcher dann zu dem normalen Rhythmus zurück.
Da Samples im allgemeinen relativ viel Speicherplatz benötigen, wurde bei dem Programm auf großen Grafik-Schnickschnack verzichtet, um somit das Chip-RAM nicht unnötig zu belasten. Die Hauptparameter werden über ein Scriptfile eingestellt. Dieses Scriptfile kann mit jedem Texteditor erstellt werden, der die Daten als normalen ASCII-Standard abspeichert. Es enthält den Pfad zu dem gewählten Sample, den Typ und seine Nummer. Die Nummer gibt dabei an, ob ein Sample als Rhythmus oder in den oben genannten Arten abgespielt wird.
Eine Besonderheit liegt in der Option des Maus-Scratchings. Sobald dieser Menüpunkt aktiviert wurde, kann die Maus als "Scratchgerät" benutzt werden. Je schneller die Maus bewegt wird, desto schneller wird auch die Abspielgeschwindigkeit eines Samples wiedergegeben.

(br)

Name: A.U.S.T.R.I.A. 21
Quelle: M.A.R. Computershop

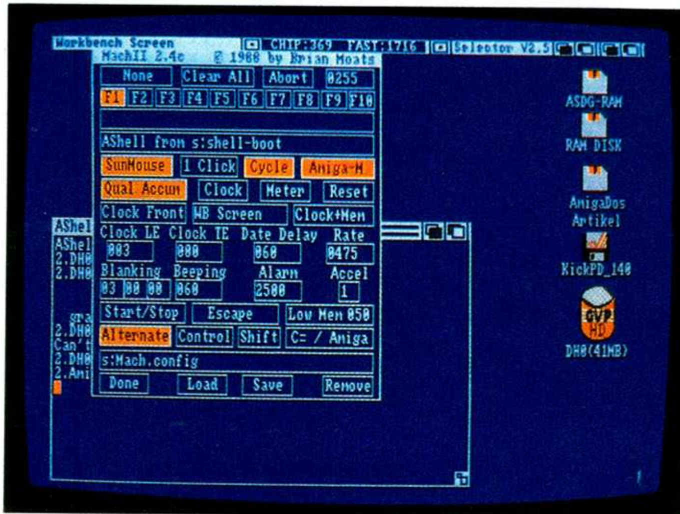
Bei der Diskette KickPD 140 handelt es sich um eine Sammlung von Hilfsprogrammen, die der Amiga-Anwender bei der täglichen Auseinandersetzung mit den Bits und Bytes gut gebrauchen kann. Schauen wir uns die Programme auf dieser Diskette einmal etwas genauer an.

KickPD 140 – alt, aber bewährt

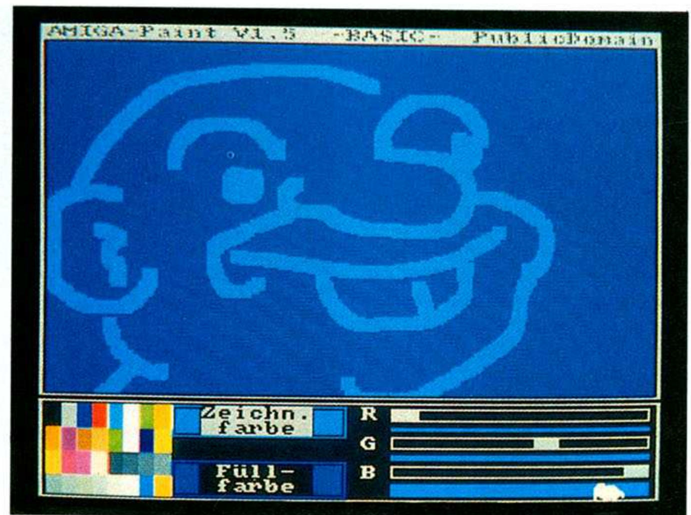
Als erstes wäre da einmal ein Programm namens **Disksalv** zu nennen. Disksalv wurde zur Rettung von zerstörten Dateien auf defekten Disketten und Festplatten geschrieben.Dieses Programm kann als Ersatz für den Amiga-DOS-Befehl Diskdoctor auf die Workbench kopiert werden. Der Vorteil von Disksalv gegenüber dem Diskdoctor ist die wesentlich größere Erfolgschance bei der Rettung des Disketteninhaltes. Der Aufruf von Disksalv ist recht einfach: Man gibt einfach das Inputdevice und das Outputdevice an

Liebe Leser,

Die Programme, auf die wir im Rahmen der PD-Seiten eingehen, finden sich größtenteils auch auf anderen PD-Reihen wieder. Die Preise für die Disketten variieren jedoch von Vertreiber zu Vertreiber. Wenn Sie sich für ein bestimmtes Programm interessieren, sollten Sie in eigenem Interesse einen Preisvergleich anstellen.



Mach II – ein Utility, das mit vielen nützlichen Features aufwarten kann



Wenn Sie ihre künstlerischen Fähigkeiten zu Papier bzw. auf den Bildschirm bringen wollen, dann ist das Programm Amiga-Paint genau das richtige für Sie

(zum Beispiel Disksalv df0: df1: oder Disksalv df0: ram:), und schon beginnt die Rettung der Daten. Disksalv arbeitet mit dem normalen Amiga-DOS und dem Fast-Filing-System zusammen. Besonders positiv ist auch die Tatsache, das Disketten von der Benutzeroberfläche aus formatiert werden können. Falls es einmal nötig sein sollte, können auch Daten, die sich auf defekten Spuren von Festplatten befinden restauriert werden.
Des weiteren befindet sich auf der Diskette **Mach II**, ein multifunktionales Tool, das in einem separaten Task läuft. Mach II bietet eine Vielzahl an Funktionen, unter anderem Screenblanker, Mausbeschleuniger, Funktionstastenbelegung und Pop-CLI-Aufruf. Die Aufzählung aller Einzelheiten würde diesen Artikel sicherlich sprengen, sie können in einer ausführlichen Anleitung, die sich auf der Diskette befindet, in Erfahrung gebracht werden. Besonders hervorzuheben ist die Tatsache, daß Mach II in die Startup-Sequence implementiert werden kann und somit sofort nach dem Booten des Rechners zur Verfügung steht. Alle Einstellungen können abgespeichert und nach dem Start automatisch nachgeladen werden.
MR-Print ist ein Utility, das zum Ausdrucken von Textfiles konzipiert wurde. Das Programm kann entweder aus dem CLI gestartet werden (dabei gibt man den Dateinamen direkt hinter dem Programmaufruf an), oder man startet MR-Print ohne Angabe des Dateinamens direkt. Das Einladen eines Textes erfolgt dann von der Benutzeroberfläche aus. Der Programmrequester, der dabei erscheint, ermöglicht eine Vielzahl an Einstellungsmöglichkeiten wie Datum an/aus, Filename an/aus, Zeilen pro Seite sowie Zeilennumerierung und vieles mehr.

Das nächste Programm ist für diejenigen, die mit 1 MByte oder mehr Speicher und nur einem Laufwerk arbeiten. Die Rede ist von **RAM-Copy**, das das Kopieren von Disketten in nur einem Durchgang erlaubt. Das Programm formatiert die Zieldiskette, beschreibt diese dann und führt zuletzt ein Verify durch. Eine Sache muß jedoch beachtet werden. Will man RAM-Copy benutzen, darf kein anderes Programm im System laufen. Erreichen kann man dies am besten dadurch, daß die Startup-Sequence mit CTRL-D abgebrochen wird, um dann RAM-Copy zu starten.
Möchte der Amiganer in den Genuß einer Zeitansage geraten, dann sollte er sich schnell das Programm **SPUD-Clock** besorgen. Dieses Programm führt in bestimmten Intervallen eine Zeitansage durch. Gestartet wird es mit dem CLI-Aufruf:

```
run SPUDClock (options)
```

Als Optionen können dabei dienen:

-f Zeitansage alle 15 Min.
-t Zeitansage alle 30 Min.
-h Zeitansage alle 60 Min.

Nach dem Aktivieren wird dem Anwender dann die aktuelle Zeit akustisch mitgeteilt.
Wer schon immer einen Taschenrechner für seinen Amiga gesucht hat, ist mit **Multicalc** sicher gut beraten. Die Geschwindigkeit von Multicalc nimmt bei sehr großen Zahlen jedoch derart stark ab, daß ein zügiges Rechnen nicht mehr möglich ist. Der Taschenrechner arbeitet nach dem RPN- (Reverse Polish Notation) System und erlaubt die meisten technisch-wissenschaftlichen Funktionen.

(Robert Münch/br)

Name: KickPD 140
Quelle: Maxon

Bei dem Programm Amiga-Paint auf der Oase 25 handelt es sich um ein (wie der Name schon ahnen läßt) Mal- und Zeichenprogramm, mit dem sich Bilder im IFF-Format erstellen lassen.

Amiga-Paint – ein komfortables Grafikprogramm

Amiga-Paint braucht sich vor kommerziellen Produkten nicht zu verstecken. Es besticht nicht allein durch eine benutzerfreundlich aufgebaute Menüsteuerung, auch findet man die meisten Standardzeichenfunktionen wieder, die auf vielen Wegen erweitert worden sind.
So kann man beispielsweise die Füllmuster in einem speziellen Untermenü editieren und so eine Vielzahl von verschiedenen Patterns erstellen und auf Diskette abspeichern. Diese lassen sich selbstverständlich auch wieder einladen und können weiter editiert werden. Das Zeichnen der Bilder wird durch viele Malfunktionen unterstützt. Es gibt drei verschiedene Pinselgrößen, Kreis-, Ellipsen-, Block-, Rahmen-, Linien-, Ray-(Strahlen-) und Füllfunktionen, es kann auch frei mit der Hand gezeichnet werden. Besondere Feinheiten können mit der Lupe nachbearbeitet werden, die unterschiedliche Zoomgrößen bereitstellt. Das erstellte Bild kann dann im IFF-Format, das mittlerweile als Standardformat angesehen wird, abgespeichert werden. Somit können auch mit anderen Malprogrammen erstellte Bilder mit Amiga-Paint eingeladen und bearbeitet werden. Last but not least kann der Anwender auf eine Hardcopyfunktion zugreifen, um seine Bilder zu Papier zu bringen.
Nebenbei bemerkt: Amiga-Paint ist in Amiga-BASIC geschrieben, worunter natürlich die Geschwindigkeit des Programms stark zu leiden hat. Alles in allem ein Programm zum Low-Cost-Preis, welches keine großen Wünsche offenläßt. Schauen Sie es sich doch einfach mal an.

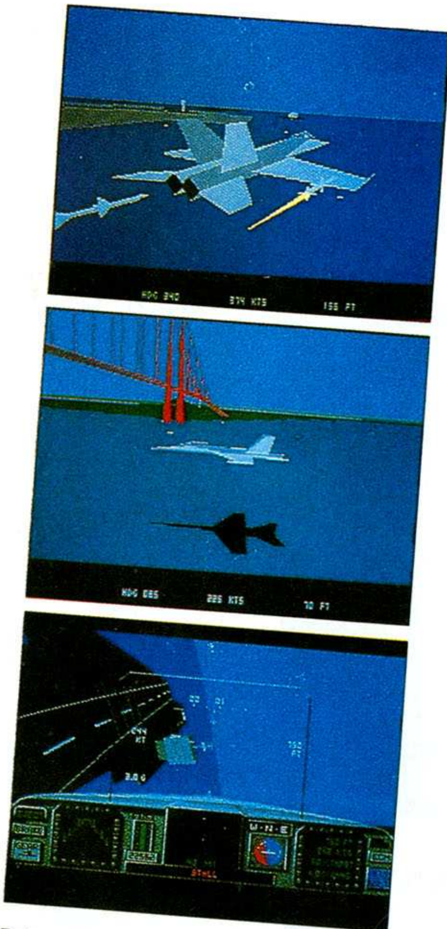
(Timo Siebert/br)

Name: Amiga-Paint
Quelle: Rainer Wolf

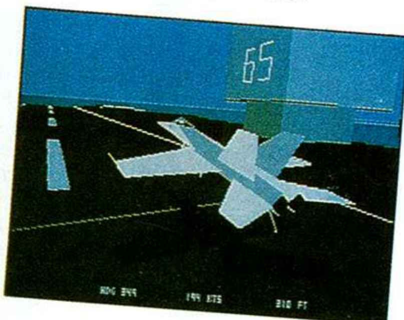
AMIGA DOS

Spielekiste

Solange der Vorrat reicht

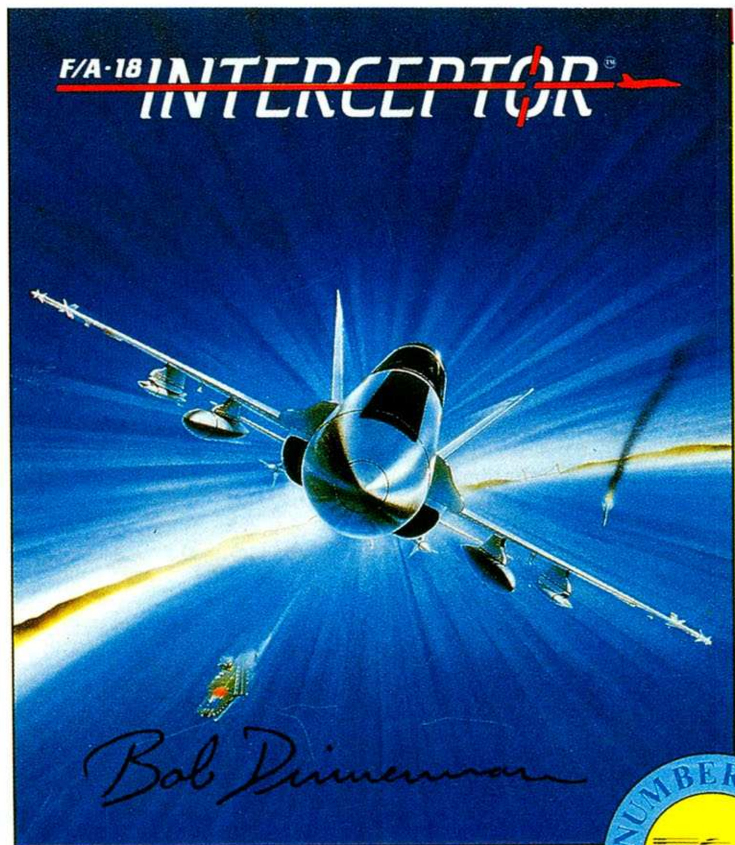


Robert Dinnerman entdeckte seine Leidenschaft für Computerspiele, als er Hardware für Motorola entwickelte. Sein Interesse führte ihn zu der Coin-Op-Abteilung von Bally, wo er an 3D-Perspektiven und Flugsimulationen arbeitete. Nachdem er einen Evans-Sutherland-Flugsimulator gesehen hatte, beschloß Bob, ein vergleichbares, aber erschwingliches Programm für Home-Computer zu entwickeln. Wenn Sie seinen F/A 18 Interceptor spielen, werden Sie sehen, daß er in der Tat etwas Außergewöhnliches vollbracht hat.



Fliegen Sie das faszinierendste Flugzeug unserer Zeit!

Fliegen Sie entweder mit der F16 oder F18, und starten Sie von einem der drei Landstützpunkte oder vom Flugzeugträger aus ● Realitätsnahe 3D-Grafik der San-Franzisko-Bucht, komplett mit Golden Gate Bridge, Alcatraz und der Pyramide in San Franzisko ● Holografisches HUD-Display und realistische Sound-Effekte ● 8 unterschiedliche Ansichten aus dem Cockpit sowie 8 unterschiedliche Außenansichten ● 7 Missionen mit unterschiedlichen Schwierigkeitsgraden.



Unser Sonderangebot an alle Amiga-Leser.
Nur solange der Vorrat reicht.

F/A 18 Interceptor, Electronic Arts,
Amiga 3,5-Zoll-Diskette — DM 39,95 *

und aus unserem laufenden Programm

DMV-Fraktalgenerator 3D Amiga
Amiga 3,5-Zoll-Diskette — DM 69,– *

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 2 50 · 3440 Eschwege

DMV Daten- und Medienverlag

Public-Domain-Spieleshow

Die qualitativen Grenzen zwischen kommerziellen Spielprogrammen und Public Domain und Shareware verwischen immer mehr. Sicherlich, die absoluten Spiele-Highlights bedürfen aufwendiger Entwicklung; PD-Programme bleiben aber, was Originalität anbetrifft, den kommerziellen Produkten nur wenig schuldig. Die Spiele, die wir für diese Ausgabe zusammengetragen haben, untermauern dies eindrucksvoll.

Roll On (V1.1) von Tobias Eckert ist eine sehr gute "Soko Ban"-Version und ist Shareware. Das bedeutet, daß der Autor nachträglich bei Verwendung des Programmes einen Geldbetrag von 10 DM verlangt. Dafür erhält der Käufer immer die neueste Version des Spieles zugesendet.

Roll On

Die Handlung des Programmes ist schnell beschrieben. Der Spieler steuert mit der Tastatur oder einem Joystick ein Männchen, welches in einem Irrgarten umherläuft und verstreut herumliegende Felsen auf Symbole schieben muß. Hat man diese Aufgabe mit allen Felsen geschafft, gelangt man in den nächsten Level.

Das Tolle an diesem Spiel ist, daß man vor jedem Schritt überlegen sollte, was man tut, ansonsten kommt man schnell in eine ausweglose Situation, indem man einen Felsen z.B. nicht mehr aus einer Ecke bekommt. Kommt man in einem Level nicht mehr weiter, hilft nur noch die DEL-Taste. Mit ihrer Hilfe kann man einen Level noch mal durchspielen.

In der Version 1.1 liegen Roll On neun Spielfelder bei. Hat man einmal alle gesehen, würde das Spiel schnell an Reiz verlieren. An dieser Stelle kommt der integrierte Level-Editor zum Zuge. Wählt man im Vorspann den Menüpunkt "Editor" an, kann man die bereits vorhandenen Spielfelder verändern oder einfach zusätzliche hinzufügen und auf Diskette abspeichern.

Hat man den Editor gestartet, kann man mit dem Joystick oder der Tastatur einen kreuzförmigen Cursor steuern. Mit verschiedenen Tastenkombinationen können an der aktuellen Cursorposition die unterschiedlichen Spielfeldelemente schnell und problemlos plaziert werden. So hat man die Möglichkeit, in kurzer Zeit für neue Motivation zu sorgen.

Name: Roll On
Enthalten auf: Kickstart 251
Vertrieb: siehe Anbieterliste

Mit Roll On erhält man ein tolles Strategie- und Geschicklichkeitsspiel in einem, das keinem AMIGA-Freund in seiner Spielesammlung fehlen sollte. Nicht zuletzt verhilft der eingebaute Editor zu viel Abwechslung und Unterhaltung.

Youpi ist genau das richtige für Computerspielefreunde, die ein buntes und lustiges Labyrinthspiel zu schätzen wissen. Dabei ist Youpi aber nicht die 1001. Pac-Man-Variante.

Youpi

Bei diesem Programm steuert der Spieler mit der rechten und linken Maustaste einen Wurm durch ein Labyrinth und muß dabei umherliegende rote Kugeln fressen. Nach jeder verzehrten Kugel wächst der Wurm um ein Glied. Die Schwierigkeit des Spieles liegt nun darin, daß man versuchen muß, alle roten Kugeln, die im Labyrinth verstreut sind, aufzufressen, ohne dem eigenen Körper dabei zu nahe zu kommen. Denn berührt der Kopf des Wurmes den eigenen Körper, verliert man ein Leben.

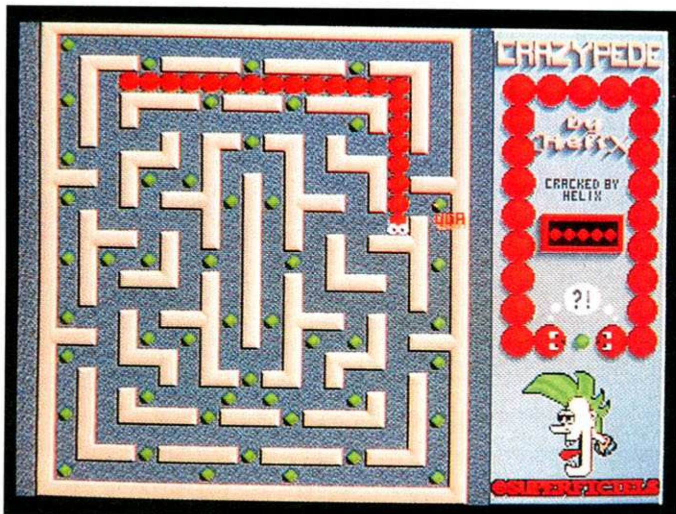
Besonders gut gelungen ist die lustige und farbenfrohe Grafik. Aber auch die Soundeffekte sind gut digitalisiert und passen zur Handlung. Ein weiterer Pluspunkt ist, daß sich optional zwei Spieler abwechselnd durch die Labyrinthe schlängeln können. Allein die Steuerung ist zu Beginn etwas gewöhnungsbedürftig.

Youpi ist ein sehenswertes Spiel.

Name: Youpi
Enthalten auf UGA PD 3
Vertrieb: siehe Anbieterliste



Roll On – es gilt Felsen an die richtigen Stellen zu schieben

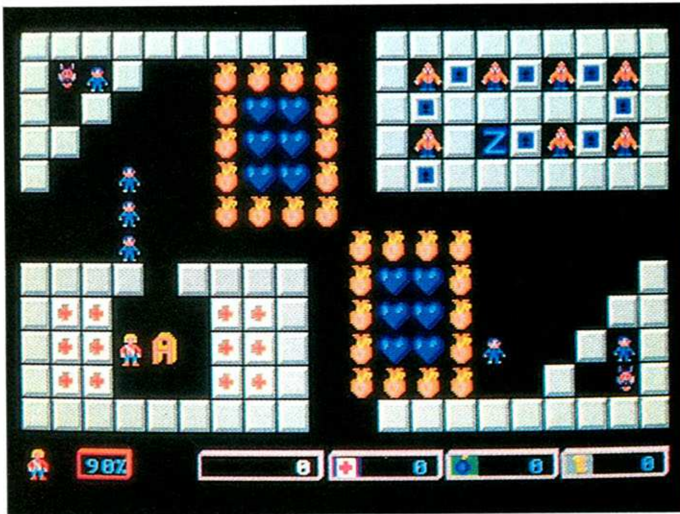


Youpi, der Wurm wird immer länger und damit schwerer zu kontrollieren.

Kennen Sie schon "Das große deutsche Ballerspiel"? Dieses Spiel macht seinem Namen alle Ehre. Der Spieler steuert dabei ein Männchen durch ein Labyrinth. Auf diesem Weg gilt es jede Menge Monster, Wizards und Gnome abzuballern .

Das große deutsche Ballerspiel

Außerdem muß man aber auch das Köpfchen einsetzen. Denn es gibt auch eine große Anzahl nützlicher Gegenstände im Labyrinth, die eingesammelt werden wollen. Auch die Wände des Irrgartens sind etwas Besonderes. Einige lassen sich einfach wegschießen und geben so den Weg in den Rücken der Monster frei. Andere Wände sind richtig gemein. Sie lassen weder das Männchen des Spielers noch dessen Schüsse hindurch, aber für Monster stellen sie kein Hindernis dar. Trifft die Figur auf eine Rotkreuz-Station, dann kann der Spieler für einen Moment aufatmen. Jedes berührte Kreuz ermöglicht dem Spieler eine einmalige Heilung, d.h. die Lebensenergie steigt wieder auf hundert Prozent an.



Ballern, ballern, ballern, DGDB ist eine sehenswerte PD-Ballerei



Zerg ist eine gut programmierte Rollenspielversion aus der Public Domain

Reichen einem Spieler die vorhandenen Level nicht aus, dann kann er sich mit einem Level-Editor einfach und schnell neue erstellen. Ist er mit den Standardmonstern grafisch nicht mehr zufrieden, dann zeichnet er sich mit dem Zeichen-Editor in wenigen Minuten einfach neue.

Die Grafik und der Sound sind nicht besonders auffallend, aber ausreichend.

DGDB(Das große deutsche Ballerspiel) ist nichts für feinfühlige Menschen. Wer aber ein Spiel sucht, das Action pur liefern soll, der ist mit DGDB richtig und gut bedient. Alle anderen Spielefans sollten die Finger davon lassen.

Name: Das große deutsche Ballerspiel
Enthalten auf Kickstart 223
Vertrieb: siehe Anbieterliste

Phantasie-Rollenspiele haben eine große Anhängerschar. Wer kennt nicht die Rollenspielserie 'Ultima'. Für viele verschiedene Computersysteme sind davon Umsetzungen programmiert worden.

Zerg

Zerg 1.0 ist der Ultima-Serie sehr ähnlich. Der Spieler schlüpft in die Rolle eines Ritters, der die Aufgabe hat, eine fremde Welt zu erforschen. Dabei stellen sich dem tapferen Ritter die verschiedensten Monster, Goblins, Orcs, Trolls und Drachen entgegen. Außerdem sind die vielfältigsten Reaktionen auf verschiedene Situationen möglich. Trifft der Spieler zum Beispiel auf eine Stadt, dann hat er die Möglichkeit, mit der Stadtbevölkerung zu sprechen, Zaubersprüche zu formulieren und vieles mehr.

Im Verlauf des Spieles findet der Abenteurer allerhand nützliche Gegenstände wie Waffen, Geld usw., die für die erfolgreiche Erforschung unabdingbar sind.

Zerg ist ungemein komplex und wird den Rollenspieler lange an der Tastatur fesseln. Dennoch ist die Bedienung nicht kompliziert geworden. Mit einfachen Tastatur-Kommandos kann das gesamte Spiel gesteuert werden. So lenkt man seinen Helden mit den Cursor-Tasten flott durch die zu entdeckende Welt oder weicht Feinden aus. Besonders wichtig ist bei einem Rollenspiel die einstimmende Anleitung. Die auf Diskette beiliegende Anleitung beschränkt sich aber auf eine einfache Bedienungsanleitung. Leider ist diese aber in englischer Sprache abgefaßt, so daß der "Rollenspieler" der Sprache auf jeden Fall in gewissen Grenzen mächtig sein sollte. Die Grafik und der Sound sind bei dieser Art von Programmen nicht besonders wichtig, dennoch sind beide durchschnittlich gut.

Name: Zerg V1.0
Enthalten auf Kickstart 252
Vertrieb: siehe Anbieterliste, Spielhallenfans aufgepaßt!



Flippern – Dungeon Flipper glänzt mit netter Grafik, guten Effekten und digitalisierten Geräuschen

Mit 'Dungeon Flipper' kann sich jeder einen sehr gut simulierten Flipper auf den heimischen Monitor holen.

Dungeon Flipper

Das Spiel bietet die Möglichkeit, bis zu vier Personen mitspielen zu lassen. Die Objekte auf dem Spielfeld, die von der Flipperkugel berührt werden können, sind grafisch gut gestaltet und bieten bei Kugelkontakten eine Menge Optionen und tolle Soundeffekte. Trifft die Kugel zum Beispiel eine Schatztruhe, dann erscheinen an den Abgängen rechts und links Goldkugeln, die diese blockieren, damit der Spieler nicht so leicht einen Ball verlieren kann.

Reagiert man einmal zu spät und der Ball fällt ins Aus, dann erzeugt der Soundchip ein tiefes, schadenfrohes Lachen.

Aber Grafik und Sound sind natürlich nicht alles. Das Programm läßt sich mit der Tastatur oder per Joystick steuern. Beide Steuermöglichkeiten sind sehr exakt und lassen schnelle Reaktionen zu.

Wer schon immer ein preiswertes Flipper-Simulations-Programm auf dem AMIGA gesucht hat, sollte sich Dungeon Flipper auf jeden Fall besorgen.

(Ingmar Reyer/hs)

Name: Dungeon Flipper
Enthalten auf Oase 108
Vertrieb: siehe Anbieterliste



R-Type – gib Aliens keine Chance

Light Force

Vier actiongeladene Spiele mit 'historischem' Wert zum Preis von einem bietet der neue Sampler von Ocean. Einsteiger sollten unbedingt einen Blick riskieren.

Wirkliche Sensationen sind auf dem Spielemarkt oft nur schwer zu entdecken. So kommt es denn, daß neuere Spiele oft nur Variationen von älteren Vorbildern sind, bei denen eine alte Idee mit meist besseren optischen und akustischen Effekten neu belebt wird. Dagegen ist an sich nichts einzuwenden, es erzeugt jedoch eine Art 'Generationenkonflikt' in der Anwenderschaft. Wer den Amiga erst seit einem Jahr nutzt, wird es vielleicht sonderbar finden, daß 'ältere Semester' bei der Erwähnung von Spielen wie R-Type, die doch mehr als zwei Jahre auf dem Buckel haben, ins Schwärmen geraten und zu nervösen Zuckungen am Feuerknopf-Finger neigen. Sampler, die in Ehren ergraute Spiele wieder neu auflegen, können hier vermitteln – zumal viele der Klassiker kaum noch zu erwerben sind. Je nach Konzept des Samplers kann man in die Urzeit der Spiele zurückgehen, wie beim an anderer Stelle bereits vorgestellten 'Game Pack' geschehen; man kann jedoch auch eine Selektion aus neueren Spielen treffen. Dies wurde in 'Light Force' perfekt praktiziert. Vier 'betagte' Spiele, von denen zumindest zwei das Prädikat 'besonders wertvoll' verdienen, wurden in einem Paket vereinigt.

R-Type ist der Urahn des modernen Ballerspiels, das nicht nur seinerzeit Aufsehen erregte, sondern auch zum Maßstab für neuere Produkte wie Xenon II, Battle Squadron oder X-Out wurde. Die Story ist simpel: Aliens bedrohen die Erde, ein Held mit seinem Raumschiff muß rettend eingreifen. In sechs farbenfrohen Levels muß alles abgeschossen werden, was sich einem in den Weg stellt. Gut animierte Gegner, flotte Grafik und guter Sound machen die Alien-Hatz zu einem wahren Vergnügen, ein breites Arsenal an Extras kann das (Über-)Leben erleichtern.

International Karate + ist ein Prügelspiel erster Güte, bei dem man den Gegner mit allerlei Tritten und Schlägen außer Gefecht setzen darf. Einfach ist dies jedoch nicht, da man gegen zwei Gegner gleichzeitig antreten muß. Diese werden entweder beide vom Computer oder einer von einem anderen Spieler gelenkt. Fetzige Musik und liebevolle Details in den Animationen und Soundeffekten machen IK+ zu einem riesigen Spielvergnügen – besonders, wenn man zu zweit 'zur Sache' geht…

Abgerundet wird der Sampler mit **Bio Challenge** , einem Geschicklichkeitsspiel mit passabler Grafik, gutem Sound und mittelprächtiger Steuerung, und **Voyager** , einem Ballerspiel mit einer durchschnittlichen Vektorgrafik und einem Geschehen im Sinne von Starglider. Die letzten beiden Spiele sind sicher keine Sensationen, aber zumindest R-Type und IK+ sollte man mal gesehen haben. Dieser Sampler bietet hierfür die ideale Gelegenheit in Sachen Preis und Leistung, die der Nachwuchs-Amigo beim Schopfe packen sollte.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Light Force
Hersteller: Ocean
Vertrieb: Fachhandel
Preis: 84,95 DM

Positiv:
- vier hochwertige Spiele zu günstigem Preis
- davon zwei 'Klassiker'
- 'historischer Wert'

Negativ:
- absolut nichts!

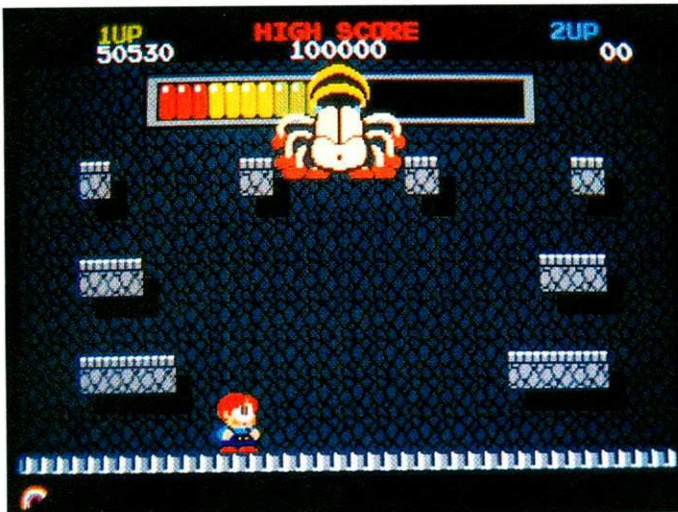




'Immer feste druff' ist die Devise von IK+



Bio Challenge schafft auch die härtesten Robots



Die Inseln des Regenbogens bieten ein recht farbiges Szenario

Rainbow Islands

So wie Sindbad seine sieben aufregenden Abenteuerreisen machte, begeben wir uns auf Erkundungszug zu den sieben Inseln des Regenbogens.

Bub und Bob, die Helden von Bubble Bobble, sind wieder auf Tour. Diesmal verschlägt es unsere Helden zu den sieben Inseln des Regenbogens, wo zahlreiche Gefahren warten. Dieses Jump'n'Run-Spiel überzeugt bereits durch gute Grafik und ein sehr farbenfrohes Szenario. Es macht damit nicht nur den großen, sondern auch den kleinen "Spielratzen" Spaß. Man hat es im ersten Level mit harmlos anmutenden Raupen, Marienkäfern und anderem Kleingetier zu tun, die dafür aber umso gefährlicher sind. Am Ende eines jeden Levels lauert der böse Wächter auf seinen großen Einsatz.

Als Waffe beziehungsweise als Hilfe benutzt man Regenbögen. Man kann an ihnen emporklettern, mit den Regenbögen auf seine Angreifer schießen oder den Regenbogen brechen, indem man auf ihn draufspringt. Dabei zertrümmert er dann alles, was auf seinem Weg liegt. Einige besonders wichtige Objekte sollte man sich nicht entgehen lassen: So kann man mit den magischen Schuhen schneller laufen, mit dem roten Zaubertrank einen weiteren Regenbogen erhalten und mit dem gelben Zaubertrank die Regenbögen beschleunigen. Natürlich warten auch in jedem Level geheime Türen darauf, daß sie geöffnet werden. Auf jeden Fall ist Eile angesagt, damit Sie nicht im wahrsten Sinne des Wortes baden gehen. Wenn Sie also Spaß daran haben, sich mit Robotern, Werwölfen und Gespenstern anzulegen, gute Grafik mögen und anhaltenden Spielspaß wünschen, ist Rainbow Islands genau das Richtige für Sie.

(vb)

AMIGA DOS Blitzlicht

Name: Rainbow Islands
Hersteller: Ocean
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- gute Grafik und Animation
- viel Spaß
- netter Sound

Negativ:
- Preis relativ hoch





Airport – nicht nur Jets stürzen ab...

Airport

Einblick in die stressige Arbeit eines Fluglotsen will diese Simulation vermitteln. Allerdings werden die Nerven des Spielers auf ganz andere Art aufgerieben.

Fluglotsen haben es nicht leicht. Unterbezahlt und ständig unter Streß müssen sie verantwortungsvoll für die Sicherheit im Luftraum sorgen. Zur Übung soll der Spieler in der Simulation für eine Schicht die Aufgaben eines Lotsen übernehmen. Flugzeuge müssen in die korrekten Luftkorridore geleitet oder sicher gestartet und gelandet werden, wozu man ihnen Flughöhe und -richtung 'per Funk' zuweist. Kontrolliert wird der Luftraum in einer Übersicht in Vogelperspektive mit Anzeige der aktuellen Lage. Damit die Arbeit nicht zu einfach wird, kommt es immer wieder zu kleineren Komplikationen. Mal müssen zufällige 'Hindernisse' wie Schlechtwetterfronten oder Vogelschwärme umflogen, mal müssen Jets mit Treibstoffmangel oder Krankheitsfällen an Bord bevorzugt behandelt werden. Aber auch die Landschaft sorgt durch Berge und Türme für Probleme.

Auf den ersten Blick sicher eine reizvolle Aufgabe, der zweite belehrt jedoch eines besseren. Im einfachen Übungsmodus kommt mit nur zwei zu kontrollierenden Jets schnell Langeweile auf, der echte Spielmodus mit seinen fünf Jets ist dagegen fast unspielbar. Dort verliert nicht nur der potentielle Fluglotse die Übersicht, sondern leider auch der Rechner: Im Test konnte jedenfalls keine Runde beendet werden, ohne daß das Programm abstürzte. Schade, daß aus einer guten Idee nicht mehr gemacht wurde, auch der einigermaßen moderate Preis hilft nicht über die Schwächen des Programms hinweg.

(Michael Anton/hs)

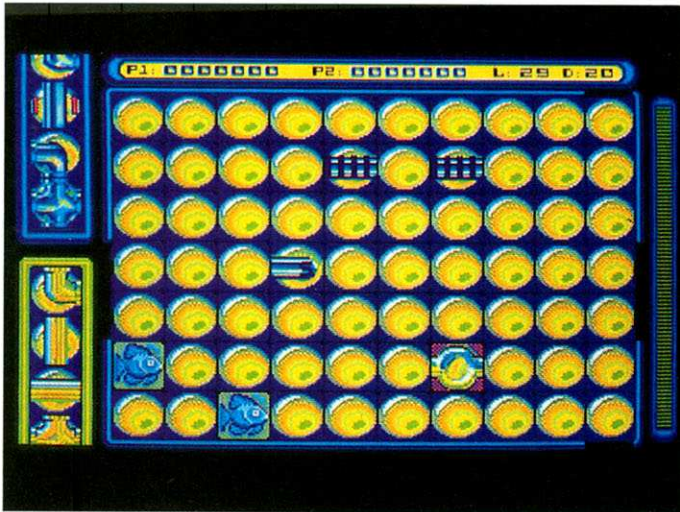
AMIGA DOS Blitzlicht

Name: Airport
Hersteller: Wolf Computertechnik
Vertrieb: Fachhandel
Preis: 49,– DM

Positiv:
- gute Idee

Negativ:
- mangelhafte Umsetzung





Beim Bauen der Pipeline kann man ganz schön ins Schwitzen kommen

Pipe Mania!!

Will man Wasser in möglichst großen Mengen möglichst schnell von einem Ort zum anderen bringen, baut man am besten eine Rohrleitung.

Bei Pipe Mania steht dem Spieler dazu ein sieben mal zehn Felder großes Spielfeld zur Verfügung. Irgendwo auf dieses Feld wird zufällig das Startstück positioniert, von welchem das Wasser nach einer gewissen Zeit losläuft.

Die Rohrstücke, aus denen die Pipeline aufgebaut wird, kommen zufällig aus einem Spender, und es muß verbaut werden, was gerade kommt. Das Ersetzen eines bereits verbauten Teils durch ein anderes kostet wertvolle Zeit und Punkte.

Kann das Wasser nicht mehr fließen, erfolgt die Auswertung des Levels: Punkte gibt es für jedes durchflossene Stück, sind genügend Teile durchflossen worden, kommt man in das nächste Level. Nach jedem fünften geschafften Level ändert sich das Aussehen des Spielfelds und man bekommt ein Passwort. In höheren Levels sind dann schon Bonusrohrstücke und Auffangbecken auf dem Feld verteilt. Außerdem gibt es dann ein Endstück, in das das Wasser zum Schluß geleitet werden muß. Pipe Mania kann auch zu zweit gespielt werden, wobei jeder Spieler seinen eigenen Teilespender bekommt. Durch Arbeitsteilung kommt man schneller durch die Levels. Pipe Mania hat ein geniales Spielprinzip, bei dem absolute Suchtgefahr besteht.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Pipe Mania
Hersteller: Empire
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- Passwortsystem
- kein Kopierschutz, daher auf Festplatte installierbar
- Zweispielermodus
- viele Extras

Negativ
- Musik wird schnell öde





Kid ist auf dem Weg nach Hause. Dabei durchquert er viele Zeiten und Orte

Kid Gloves

Kid hat sich heimlich die Boxhandschuhe seines Onkels angezogen. Diese besitzen aber magische Kräfte, die ihn in einen exotischen Urwald teleportieren. Kid muß sich also auf den sehr beschwerlichen und vor allem gefährlichen Heimweg machen.

Kid Gloves ist ein Plattformspiel. Kid versucht, vom Spieler gesteuert, springend und laufend ein Bild nach dem anderen zu schaffen. Auf den Plattformen verstreut liegen viele nützliche Gegenstände herum: Geld, das in Shops gebraucht wird, Früchte und Eiskrem, die Punkte bringen, oder Schlüssel, mit denen sich Hindernisse aus dem Weg räumen lassen. Damit das Spiel nicht ganz so leicht ist, haben die Programmierer in jedes Bild eine Vielzahl von Monstern gesetzt. Wehren kann Kid sich zunächst nur mit Hilfe einer Münze, die er als Geschoß einsetzt. Ab und zu taucht ein Shop auf, in dem Kid seine Ausrüstung nach Belieben vervollständigen kann. Bezahlen tut er dabei mit dem aufgesammelten Geld. Hier kann er zum Beispiel bessere Waffen oder zusätzliche Schlüssel erstehen. Kid kann auch Zaubersprüche einsetzen, jedoch beherrscht er diese Kunst noch nicht richtig, und so ist es zufällig, welchen Zauber er ausspricht. Alle zehn Bilder wechselt Kid in eine andere Zeit, bis er schließlich nach Bild fünzig wieder daheim ist. Die Bilder hängen teilweise zusammen. Es kommt nicht selten vor, daß man drei Bilder zurücklaufen muß, um ein Bild weiter vorn zu schaffen. Kid Gloves bietet viel Abwechslung, auch wenn an manchen Stellen, vor allem in den höheren Levels, schnell Frust aufkommen kann.

(Robert Marz/hs)

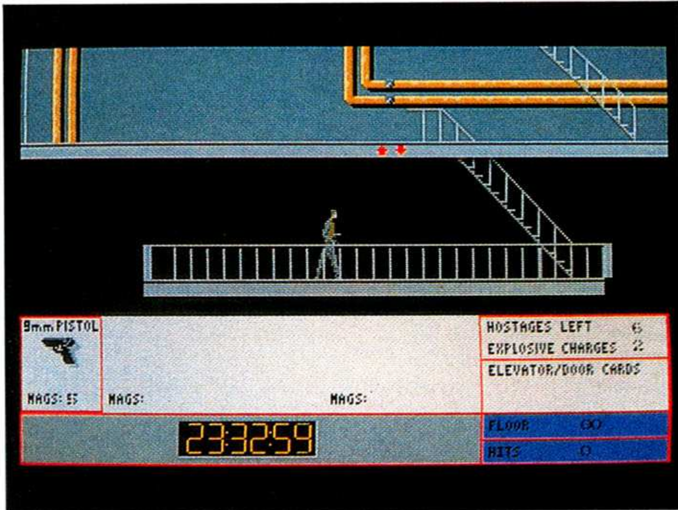
AMIGA DOS Blitzlicht

Name: Kid Gloves
Hersteller: Millenium
Quelle: Fachhandel
Preis: 84,90 DM

Positiv:
- Anleitung deutsch
- viel Abwechslung

Negativ:
- an manchen Stellen leichte Frustgefahr
- keine Speicheroption





Nur tote Terroristen sind gute Terroristen

North Sea Inferno

Weg mit den Skrupeln und her mit dem Joystick! Statt auf abstrakte Aliens darf auf 'echte' Menschen geschossen werden…

Terroristen haben eine Ölplattform nebst einigen Geiseln in ihrer Gewalt und drohen darüber hinaus mit der Zündung einer Atombombe. Dies darf natürlich nicht sein, also muß ein Retter her. Der rennt durch das Labyrinth der Plattform und schießt auf alles, was ihm in die Quere kommt. Hinter einigen Türen findet er Waffen und Munition, hinter anderen die zu befreienden Geiseln. Einige Türen und Aufzüge öffnen sich erst mit den Keycards, die ebenfalls zu suchen sind, manchmal findet er auch Verbandszeug, mit dem Verletzungen kuriert werden. Wird die Aufgabe nicht innerhalb von 30 Minuten erledigt oder steckt der Held mehr als vier Treffer ein, ist das Spiel gelaufen.

Das Thema ist sicher nicht jedermanns Geschmack, wer sich jedoch gerne als Amiga-Rambo betätigen möchte, kann dies hier tun. Hohe Anforderungen werden nicht gestellt: Den Plan der Plattform hat man schnell im Kopf, ebenso die Lage der wichtigen Räume. Auch gibt es kaum echte Überraschungen, denn die Gegner kündigen sich durch lautes Geschrei an – solange genug Munition vorhanden ist, kann kaum etwas passieren. Allerdings sind die Jungs doch etwas hart: Manchmal laufen und schießen sie durch Mauern, ein andermal schweben sie mitten in der Luft – soviel zum Thema Programmierung. Der Rest des Spiels ist kaum der Rede wert. Am gelungensten sind noch die Soundeffekte, die Hintergrundmusik nervt dezent. Die Grafik ist, von den Anfangs- und Endbildern abgesehen, 'geradlinig' und einfach, die Sprites sind extrem mager. (Immerhin sieht man das Blut spritzen…) Alles in allem ein simples Spiel ohne besondere Highlights.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: North Sea Inferno
Hersteller: Magic Bytes
Vertrieb: Fachhandel
Preis: 44,95 DM

Positiv:
- gute Soundeffekte

Negativ:
- zu wenig Variation





688 Attack Sub – unter Wasser ist die Welt nicht immer in Ordnung

688 ATTACK SUB

Auf dem PC hat diese U-Boot-Simulation bereits für Furore gesorgt. Nun ist sie auch für den Amiga verfügbar. Lohnt es sich?

Willkommen an Bord eines der modernsten U-Boote der amerikanischen Marine. Zehn Missionen warten darauf, erfüllt zu werden. Neben einfachen Übungen müssen auch Aufgaben ausgeführt werden, die taktisches und strategisches Fingerspitzengefühl notwendig machen. Torpedos auf Übungsziele zu feuern ist nur der Anfang, erst wenn alle Feinheiten des U-Boot-Krieges beherrscht werden, können Sie sich einen wahren Meister nennen. Taktik und Strategie sind gefragt, will man sich zu einem echten Seebären entwickeln. Die Wahl der Seiten ist weitgehend frei, wer der amerikanischen Technik überdrüssig ist, darf auch auf einem sowjetischen U-Boot anheuern und dort ähnliche Aufgaben übernehmen. Kampf und Navigation liegen in Ihren Händen, von Ihren Entscheidungen hängt der Erfolg der Missionen ab.

Insgesamt bringt 688 Attack Sub ein gutes U-Boot-Feeling, nicht immer realistisch, aber in der Kombination aus Action und Strategie gut durchdacht. Technische Mängel müssen leider konstatiert werden, weil der Sound oft schlecht programmiert wurde. Die gesampelten Effekte sind zwar nicht zu verachten, dennoch oft unsauber und den Ablauf störend. Am schmerzlichsten ist jedoch, daß in der Amiga-Version auf die Zwei-Spieler-Option via Modem verzichtet wurde. Dennoch ist 688 ATTACK SUB eine sehenswerte Simulation für den Amiga.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: 688 Attack Sub
Hersteller: Electronic Arts
Vertrieb: Fachhandel
Preis: ca. 89 DM

Positiv:
- gelungene Simulation
- anwenderfreundlicher Kopierschutz
- deutsche Anleitung

Negativ:
- lieblose Umsetzung
- schlechter Sound



AMIGA DOS -Spieletips

Auch wenn viele nun in der Sonne liegen und den Computer Computer sein lassen, Zeit zu Computerspielen ist jederzeit. So lassen andere Leute den Rollo herunter, sperren die Sonne aus und spielen trotzdem. Und alle, die nun nach Tips zu schweren Spielen suchen, kommen voll auf ihre Kosten. Dieses Mal erwarten Sie neben ausgiebigen Tips zu Sierra-Spielen noch eine ganze Menge Minitips und Karten zu Infestation.
Hier nun der Rest der in der letzten Ausgabe begonnenen Ausführungen über Larry Laffers zweites Abenteuer.

Die Insel, an der wir stranden, birgt ihre Gefahren, was wir spätestens dann bemerken, wenn wir rechts und links auf lauernde KGB-Agenten stoßen. Also ab durch die Mitte und hinein in den Dschungel.

Eine paradiesische Insel

Die dortige Pflanze nehmen wir im geeigneten Moment mit, aus dem Restaurant das Messer und aus dem Zimmer Seife und Streichhölzer. Der örtliche Barbier verpaßt uns eine ganz spezielle Frisur, die uns auf eine Idee bringt. Mit dem Bikini-Unterteil vom hiesigen FKK-Strand, dem Top vom Schiff, etwas Seife, mit der wir das Top ausstopfen und der Frisur sind wir so weit getarnt, daß wir an den Agenten vorbei kommen.

Auf dem Flugplatz

Am Ende der Kletterstrecke schlüpfen wir wieder in unseren Leisure Suit und nähern uns dem Flughafen. An den beiden am Eingang kommt man vorbei, indem man ihnen die Blume gibt. Der hiesige Frisör verschafft uns wieder Ähnlichkeit mit unserem Paßbild und schenkt uns ein Haarwuchsmittel. Da die Schlangen vor den Schaltern viel zu lang sind, sehen wir uns in den Räumlichkeiten einmal um. Durch den Paß erlangen wir Zugang zur Gepäckkontrolle und anderen Räumlichkeiten. Aus dem Automaten bei der Bar holen wir uns eine Versicherung und bestellen uns ein Essen. Statt es zu essen, nehmen wir es und gelangen auf diese Weise in den Besitz eines Bobby-Pins. Wir schauen uns die Koffer auf dem Fließband genauer an (get suitcase). Auf diese Weise gelangen wir dann doch noch an ein Ticket und gehen zu Gate 1. Bevor wir das Flugzeug betreten, nehmen wir noch ein Flugblatt (Pamphlet).

Im Flugzeug

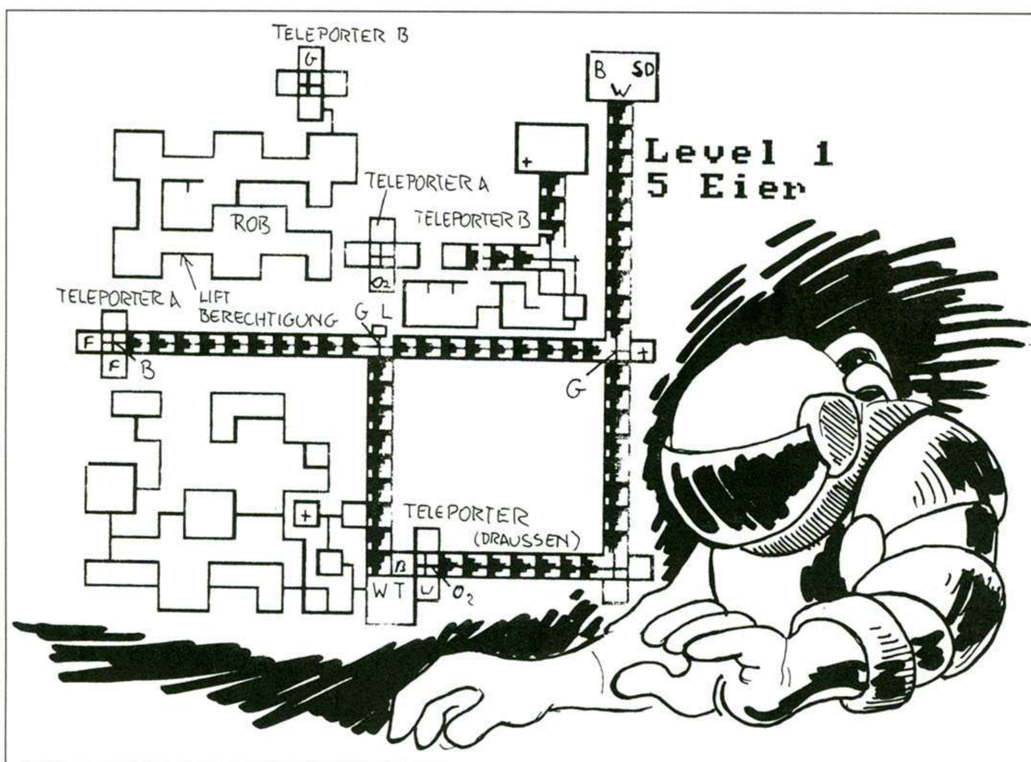
Um den beiden Typen, die uns bei der Landung in Empfang nehmen wollen, zu entgehen, müssen wir etwas tun. Wir nehmen also den Airsick Bag und versorgen den Sitznachbarn mit etwas Lektüre (das Pamphlet). Wir begeben uns zum Notausgang im Heck des Flugzeuges. Das Schloß wird mit dem Bobby-Pin geöffnet, der Hebel bewegt und die Tür geöffnet. Um den drohenden Aufprall zu vermeiden, müssen wir den Fallschirm nicht nur anziehen, sondern auch noch öffnen.

Nach der Landung

Nachdem wir uns losgeschnitten haben, verlieren wir beim Sturz das Onklunk, ohne je dessen Geheimnis erfahren zu haben. Den Stout Stick nehmen wir für den späteren Gebrauch mit. Unter dem Busch mit den Killerbienen können wir unbescholten durchkriechen (crawl), und die Schlange können wir mit dem Stock erledigen. Im Sumpf brauchen wir dann noch etwas Geschicklichkeit, ebenso wie bei der Überquerung des Flusses. Wir spielen Tarzan (jump vine) und nehmen nach der Landung eine dieser Ranken mit.

Das Finale

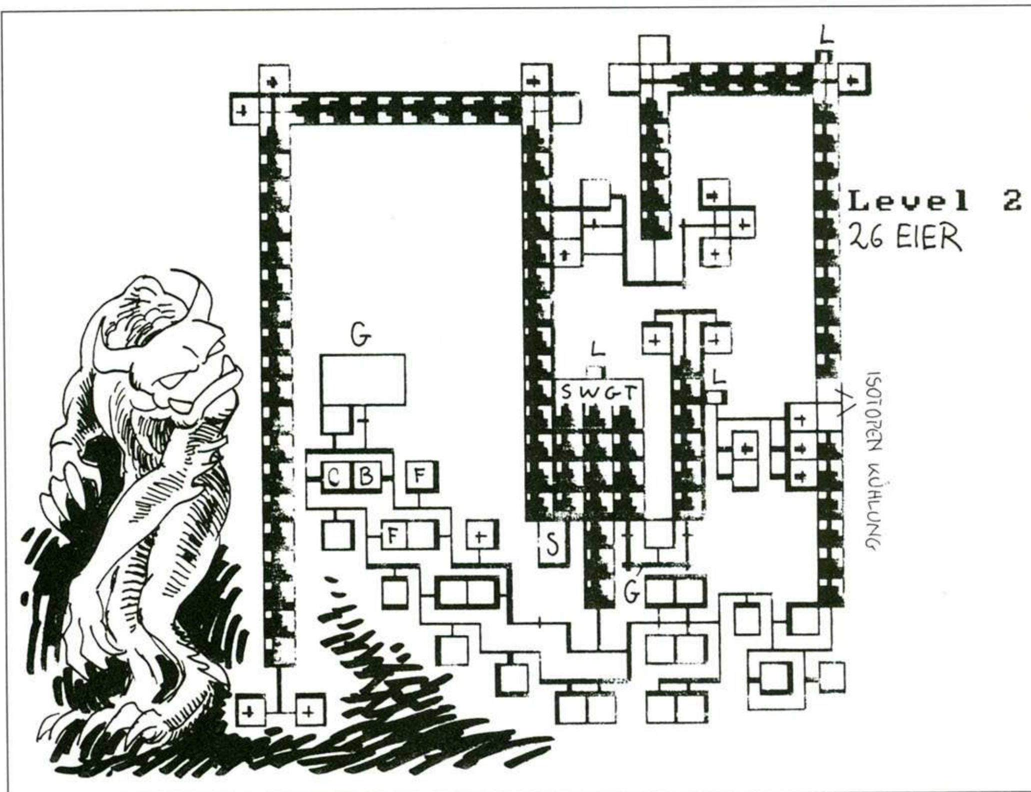
Die Klippe meistern wir mit der Liane und den Gletscher können wir mit Sand aus der Lagune oder Asche vom Dorfplatz begehbar machen. Auf der Spitze des Berges veranstalten wir an der richtigen Stelle ein kleines Feuerwerk. Die Stelle ist leicht gefunden, man muß sich allerdings Pixel-genau plazieren. Dann basteln wir uns einen Molotowcocktail, indem wir den Airsick Bag in die Hair-Rejuvenator-Flasche stecken und mit den Streichhölzern anzünden. Wenn wir jetzt die Flasche werfen, öffnet sich eine Aufzugstür, und wir haben es geschafft.
In dieser Lösung haben wir nur die nötigsten Tips zum Spielen gegeben. Wer seine fünfhundert Punkte vollkriegen möchte, muß sich selbst noch ein wenig umsehen, was ja am meisten Spaß macht. Aus diesen Gründen haben wir auch auf eine Schritt-für-Schritt-Lösung verzichtet, um Ihnen möglichst viel Spielraum zu lassen.
Weiter geht's mit Sierra-Tips. In den Tiefen des Weltraums folgen wir den Spuren des kosmischen Straßenfegers Roger Wilco.



Infestation-Karte, Level 1

Space Quest II

Zu Beginn können wir Roger, also uns selbst, beim Deckschrubben zusehen. Abgelenkt von den Signalen unserer Armbanduhr, lassen wir den Besen sausen und betrachten sie. Was die Stunde geschlagen hat, erfahren wir durch Druck auf die entsprechenden Tasten. Da wir uns unserem Chef nicht widersetzen wollen, begeben wir uns in die Schleuse. Hier ziehen wir uns zunächst einmal um, um dem Chef nicht im Raumanzug unter die Augen treten zu müssen, was dieser auch sehr übel nehmen würde. Daß der Schrank auf jeden Fall geplündert wird, braucht nicht extra erwähnt zu werden. Jetzt können wir zum Boß, der uns zu neuer Arbeit einteilt. An den neuen Arbeitsplatz gelangen wir mit der Rohrbahn, die sich im hinteren Teil des Bildes befindet. Um unseren Job erledigen zu können, bleibt uns nichts anderes übrig, als den Shuttle zu betreten. Von jetzt an können wir uns getrost zurücklehnen und die kommenden Ereignisse mit dem Drücken der Return-Taste zur Kenntnis nehmen.



Infestation-Karte, Level 2

Willkommmen auf Labion

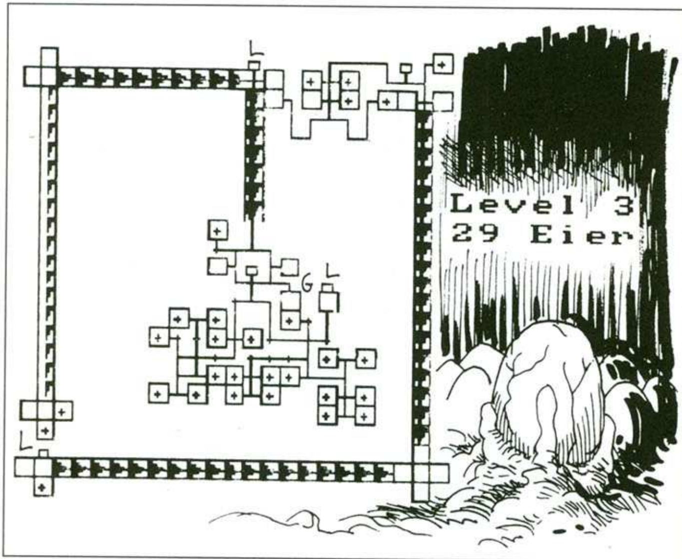
Wenn wir am Ende der Sequenz auf den harten Boden der Tatsachen zurückgeholt werden, schauen wir uns um und beschaffen uns die Keycard, die der Wächter bei sich hat. Da es ihn wohl kaum stören wird, brauchen wir auch keine moralischen Bedenken haben. Der Piepton, der so fürchterlich nervt, kann und soll mit einem Druck auf den Knopf im Inneren des Wracks abgeschaltet werden, da er zum Ortungsgerät der Plattform gehört. An dieser Stelle sollten wir ruhig einmal zwischenspeichern. In den nächsten Bildern lauern einige exquisite Todesarten: Beim Verlassen des Bilds nach rechts kann man leicht in eine Fallgrube stürzen und die mutierten Champignons im nächsten Bild würden einen Happen Roger Wilco sicher nicht verschmähen.

Vom Bild mit der Plattform schlagen wir uns ins Gebüsch und lauschen erst einmal. Die Ursache des solchermaßen registrierten Geräusches entpuppt sich als kleines Männlein, das in eine Falle geraten ist. Da es uns später noch gute Dienste leisten wird, sollten wir es befreien, indem wir uns unter es stellen und mittels 'free man' aus seiner mißlichen Lage erlösen. Zurück ins Bild davor. Hier empfiehlt es sich, einen sicheren Punkt in der Mitte des Bildes aufzusuchen, um den rabiaten Methoden des Suchtrupps, der bald erscheinen wird, zu entgehen. Dazu müssen wir uns so hinter dem Baum plazieren, daß nichts mehr von uns zu sehen ist. Die richtige Stelle wird auch mit fünf Punkten honoriert. Ist der Suchtrupp fort, folgen wir der Steigung und wechseln links oben in das nächste Bild. Dort steht ein Briefkasten im Urwald, mit dessen Hilfe wir unsere Bestellung abschicken und eine praktische Pfeife erhalten, die wir später noch brauchen werden. Im unteren Teil des Bilds geht es weiter. Um dorthin zu gelangen benutzen wir die Abkürzung über die Klippe, denn es ist einer der wenigen Stürze, die wir überleben. Unten versorgen wir uns mit einer Spore. Dabei ist Pixel-genaues Arbeiten vonnöten, da die Pflanze bei Berührung ihre Sporen auf uns losläßt und Roger diese Aktion mit einer längeren Lähmung quittiert.

Wir verlassen das Bild durch den mittleren Ausgang am hinteren Bildrand. Wenn wir vorher freundlich waren, zeigt uns das kleine Männchen, was wir tun müssen: nämlich hinten an den Sträuchern Beeren sammeln. Dazu müssen wir aber an der Spinnenpflanze vorbei, die uns den Weg versperrt. Wir müssen also durch die ausgelegten Fangarme hindurch. Was es heißt, die Arme zu berühren, wird Roger sehr schnell und auf unangenehme Art und Weise erfahren. Es ist also sehr ratsam, jeden erkämpften Pixel abzuspeichern. Hinten angekommen, pflücken wir ein paar Beeren und stellen fest, daß uns der Weg, den das Männchen genommen hat, verwehrt ist. Es hilft nichts, wir müssen wieder zurück. Wir gehen über die Mitte vorn und Mitte rechts auf die bereits wohlbekannte Lichtung, die diesmal hinten rechts verlassen wird. Da es keinen anderen Weg mehr gibt, müssen wir durch den Sumpf. Nach einigen erfolglosen Versuchen, sollten wir doch den erneuten Hinweis des Männchens beachten und uns mit den Beeren einreiben, worauf uns das Krokodil angewidert wieder ausspucken wird (rub berries on skin). Den Sumpf durchqueren wir am besten am oberen Bildschirmrand. Sobald wir im folgenden Bild den Boden unter den Füßen verlieren, sollten wir das Luftanhalten nicht vergessen (hold breath) und die Unterwasserwelt einmal näher erkunden. Auf diese Weise finden wir ein leuchtendes Juwel, das später für klare Sicht sorgt. Nach diesem kleinen Tauchexkurs gehen wir weiter nach rechts bis zu dem hohlen Baum, der, einmal erklommen, eine wichtige Brücke im Adventure bildet. Einmal drüben angekommen, geht es weiter, bis wir in eine Falle treten. Nach zwei Kommunikationsversuchen kommt uns unser Peiniger etwas näher, so daß wir ihn mit der Spore ins Jenseits befördern können. Durchsuchen wir ihn, finden wir einen Schlüssel, mit dem unserer Befreiung nichts mehr im Wege steht. Das Seil wandert in unsere Tasche, und wir gehen durch Mitte nach hinten ab. Nach einigen (tödlichen) Belehrungsversuchen durch empörte Wächter wenden wir uns nach links vorne und finden uns an einer altbekannten Stelle wieder.

Der Weg nach oben führt nach unten

Das Seil kommt nun zu seinem Einsatz, indem wir uns



Infestation-Karte, Level 3

diskret abseilen (tie rope to log; climb rope). Den Vetter von Godzilla setzen wir durch einige Manöver außer Gefecht, um deren Brillanz uns selbst Tarzan beneidet hätte. In der Finsternis der Höhle bringen wir mit dem Juwel Licht ins Dunkel (hold gem), was jedoch den Absturz, der nun folgt, auch nicht verhindern kann. Unten angekommen, rächt es sich spätestens jetzt, daß wir dem Männchen doch nicht geholfen haben. Ansonsten brauchen wir nur das Wort auszusprechen und es kommt uns zu Hilfe. Welches Wort fragen Sie? Nun eben das Wort (say the word), worauf sich völlig neue Perspektiven und Gänge eröffnen. Wenn es im folgenden wieder etwas dunkel wird, liegt es vor allem daran, daß das Juwel immer noch da liegt, wo wir es nach unserem Sturz verloren haben. Also sollten wir es wohl aufheben. Da aber beide Hände voll sind, stecken wir es in einen der weniger belegten Plätze (put gem into mouth). Nachdem wir uns wieder orientieren können, stellen wir fest, daß der Weg nach rechts unten führt. Ein Exkurs nach rechts oben erlaubt das Kennenlernen eines weiteren freundlichen Bewohners von Labion. Aber zurück zum richtigen Weg. In der Höhle angekommen, halten wir uns weiter rechts, bis zur letzten Abzweigung. Das Ende dieses Weges ist zwar ein wenig überraschend, aber durchaus ohne negative Nachwirkungen. Frisch gewaschen halten wir uns weiter rechts, bis es nirgends mehr weitergeht.

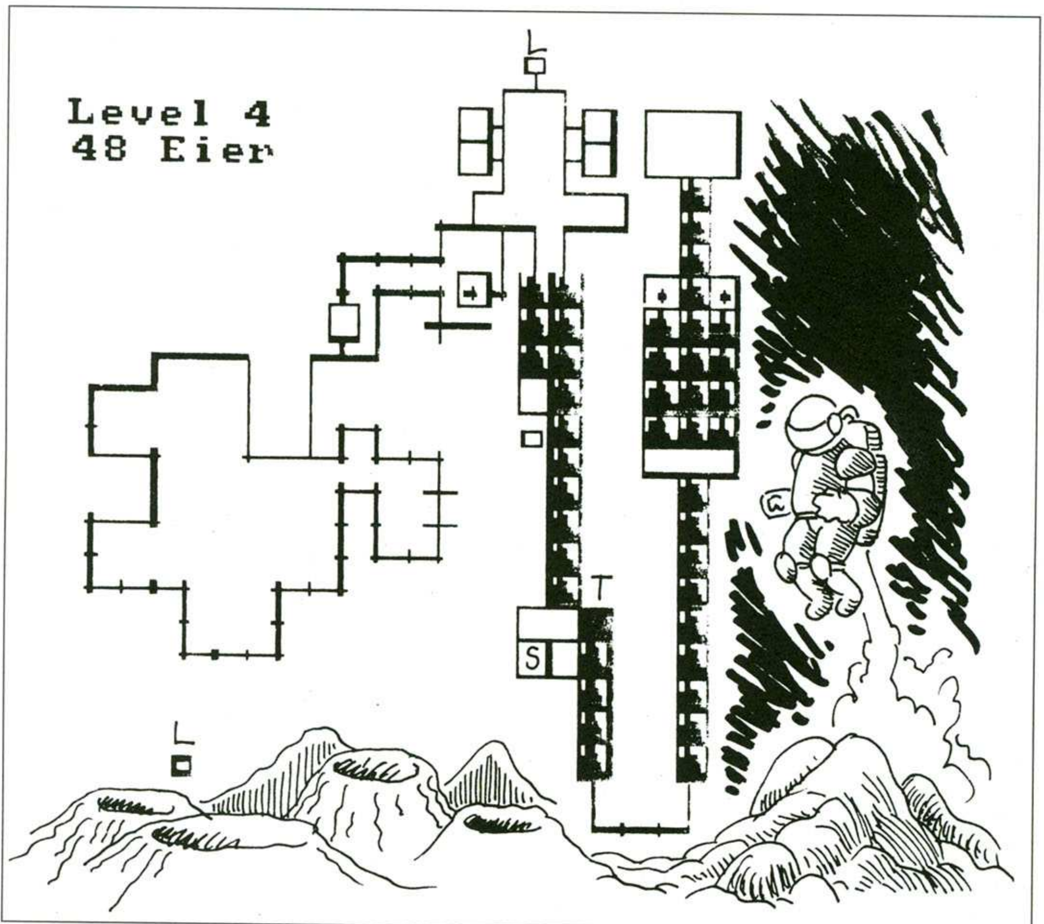
Jetzt kommt die Pfeife zur Geltung. Das daraufhin erscheinende 'Krümelmonster' läßt sich mit dem Zauberwürfel beschäftigen.

Allerdings sollte man, frei nach Goethe, die Geister lieber aus einem gewissem Sicherheitsabstand rufen.

Da das Beast von Rubik's Cube ziemlich gefesselt ist, kommen wir problemlos an ihm vorbei. In der Passage nehmen wir ein Steinchen mit, mit dessen Hilfe wir den Wachposten außer Gefecht setzen können (sling stone at guard).

Den Sternen entgegen

Die Keycard verschafft uns Zugang zur Shuttle-Plattform. Das startbereite Shuttle bekommen wir mit der uns eigenen Intuition (nicht die Benutzeroberfläche des Betriebssystems!) schnell in den Griff und verlassen souverän den ungastlichen Planeten. Die Freude darüber währt allerdings nur kurz, denn schon bald zieht uns unser Gegenspieler Sludge Vohaul in seinen Bann. In seiner Asteroidenfestung angekommen, verlassen wir das Bild unten in der Mitte und stellen fest, daß wir nicht weiterkommen. Es sind allerdings einige Hilfsmittel in dem Asteroiden versteckt, die wir nur noch zu finden brauchen. Oberhalb der Plattform existieren einige Gänge, die durch Aufzüge links und rechts davon zu betreten sind. Damit das nicht zu einfach wird, ist das System wie alles in diesem Spiel etwas merkwürdig. Es ist empfehlenswert, das oberste Stockwerk nur von der linken Seite aus zu betreten. Tun wir das nicht, steht die Konfrontation mit einem ziemlich anschmiegsamen Wesen ins Haus. Nehmen wir seinen Kuß entgegen, hat das keine unmittelbaren Folgen. Kurz vor Ende des Spiels werden wir aber am eigenen Leibe Zeuge einer Verwandlung, die Kinogängern aus den beiden Alien-Filmen bekannt sein dürfte. Also betreten wir zunächst den rechten Aufzug, fahren bis in den fünften Stock und gehen dort nach rechts, bis wir auf eine Tür stoßen. Diese öffnet sich auf Knopfdruck. Neben einem Korb finden wir beim Durchsuchen einiger Overalls ein Feuerzeug. Die Erkundung der Beschaffenheit eines nahegelegenen Fellbündels hat amüsante aber tödliche Nebenwirkungen. Lassen wir das also und begeben uns nach links, zurück in den Lift und betreten den vierten Stock. Wir folgen dem dorti-



Infestation-Karte, Level 4

gen Gang, bis wir auf zwei Türen stoßen. Es ist egal, durch welche wir eintreten. In der dortigen Bedürfnisanstalt sammeln wir etwas Papier. In die gleiche Richtung weitergehend, können wir in einem anderen Raum einen Seitenschneider auftreiben. Kurz danach erreichen wir den Aufzug am anderen Ende des Gangs. Dieser bringt uns einen Stock tiefer. Dort finden wir hinter der Tür einen Plunger, ein Gerät, welches vor allem Klempnern bekannt sein dürfte. Da es keinen zweiten Stock gibt, finden wir uns sehr schnell am Ausgangspunkt wieder. Jetzt können wir endlich durch die Mitte nach unten. Die Falle überwinden wir mittels genauen Timings und des Plungers (put plunger on wall) – aber erst ganz spät. Hat sich der Boden jetzt wieder geschlossen, lassen wir den Plunger los (release grip) und erforschen die nähere Umgebung. Was wir dabei sehen, erfreut uns allerdings gar nicht: Die fiesen Wallbots setzen wir aber mit einem gezieltem Schauer außer Gefecht. Ein Blick an die Decke offenbart nämlich eine Sprinkleranlage. Mit dem in den Korb gestopften und an der richtigen Stelle plazierten Papier und dem Feuerzeug machen wir ein kleines Feuerchen, welches den Sprinkler in und die Wallbots außer Betrieb setzt.

Showdown

Nachdem diese Gefahren beseitigt sind, begeben wir uns frohen Mutes nach rechts in die geheimen Hallen des üblen Vohauls. Dieser wartet schon auf uns und verkleinert uns mit einer teuflischen Maschine auf ein Mindestmaß. Das einzige, was wir gerade noch in Augenschein nehmen können, sind seine greulichen Versicherungsfritzen, ein Ansporn für den weiteren Kampf. Mittels des Glasschneiders befreien wir uns aus der Käseglocke, in der wir eingesperrt sind. Da wir vorne links von 'hilfreicher Hand' auf den Boden der Tatsachen zurückgeholt werden, erforschen wir die Lüftungskanäle links in der Mitte und geraten dabei (enter vent) in den technischen Mittelpunkt von Vohauls Herzschrittmacher, den wir mit einem beherzten Knopfdruck ausschalten. Nun brauchen wir nur noch dafür zu sorgen, daß wir wieder unsere richtige Größe erreichen (pull switch; type enlarge), und unser Kampf kann weitergehen. Intensive Suche zeigt uns den Code, mit dem wir die Invasion der Versicherungsvertreter per Eingabe am Terminal (look screen) aufhalten können. Ist dies getan, brauchen wir nur noch unsere eigene Haut zu retten.

Flucht

Die Treppe hinten rechts führt uns zu einer Passage durch den Asteroiden – die an ihrem Anfang angebrachte Sauerstoffmaske sollten wir nicht nur mitnehmen, sondern auch aufziehen! Die Passage führt uns zu einem Rundgang, der schon vom Shuttle aus sicht-, aber nicht erreichbar war. Um nun per Escape-Pod aus dem Chaos zu entkommen, sind mehrere (wortwörtlich!) Anläufe nötig...

Nachdem wir Roger jetzt auch bei seinem zweiten Abenteuer begleitet haben, wollen wir uns zunächst einigen anderen Spielen zuwenden. Patrik Klees aus Oetwil a.d.L. in der Schweiz sandte uns gleich einen ganzen Haufen Tips ein. Einige davon wollen wir diesmal veröffentlichen:

Das Weltraum-As

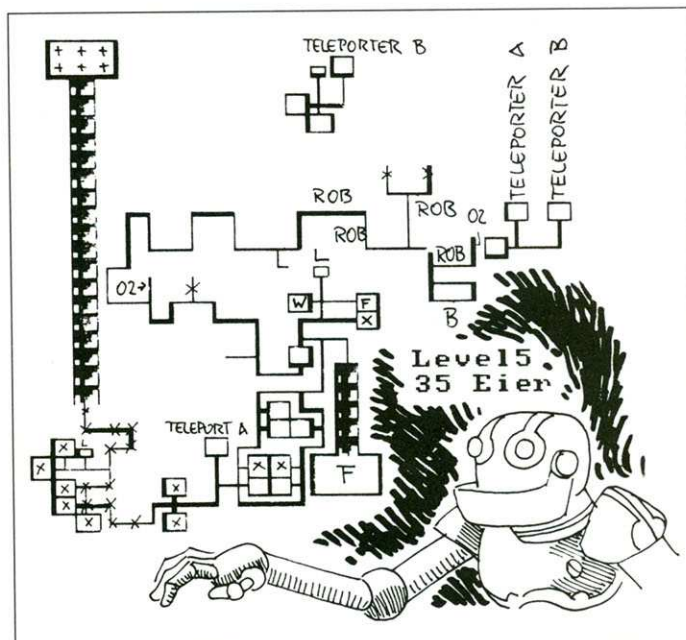
Bei dem Dragons-Lair-Nachfolger Space Ace kann man sich mit der Eingabe von 'DO DEMO DEXTER', abgeschlossen mit der Return-Taste, ein Demo des Spiels ansehen.

R-Type

Bei diesem sehr schweren Ballerspiel kann man sich die hinteren Levels ansehen, wenn man sich in der Highscore-Liste mit dem Namen 'SUMITA." einträgt (den Punkt nicht vergessen!). Auf diese Weise erhält man nämlich unendlich viele Raumschiffe.

Bomzuball

Einige der Passwörter lauten:
RATT
DAVE
IRON
WEED
LEAD
RING
GOLD
SONG
FIRE
TREE
LAMP



Infestation-Karte, Level 5

Ebenfalls einige Passwörter zu Pipe Mania haben wir beim Testen herausgefunden. Da es, obwohl wir etliche Levels weitergespielt haben, keine neuen Passwörter mehr gab, gehen wir davon aus, daß es alle sind.

Gravity Force

Christian Kristukat aus Lindau hat uns das letzte und beste Passwort zu Gravity Force zugesandt. Gibt man als Passwort WARPxx ein, kommt man in den gewünchten Level, wenn man für xx die Levelnummer setzt.

Frank Plohman hat einen tollen Trick für das Ballerspiel X-Out.

X-Out

Wenn man 500000 Credits bekommen will, braucht man sich nur Schiff Nummer vier zu kaufen und es auf dem Raster abzulegen. Nimmt man jetzt einen normalen Satelliten, legt ihn auf die Anzeige der Credits und drückt auf den Knopf, ist man reicher und kann sich eine tolle Ausrüstung kaufen.

Ebenfalls gleich mehrere Tips sandte uns Stephan Schumacher aus Uetze zu. Zunächst führt er einen Cheat zu Battle Squadron an.

Schlacht-Schwadronen

Drückt man im Hauptmenü die Space-Taste, gelangt man in ein weiteres Menü, in dem man Änderungen wie zum Beispiel die Anzahl der Leben, Geschwindigkeit der gegnerischen Kugeln und so weiter vornehmen kann. Tippt man während des Spiels das Wort CASTOR ein, ist man von da ab unverwundbar. Mit den Tasten F1 – F6 kann man die Schußstärken einstellen, mit F7 – F10 die Art der Schüsse wählen.

Die 'M's, die man einsammeln kann, werden ausgelöst, wenn man den Feuerknopf drückt und den Joystick einmal entgegen den Uhrzeigersinn dreht.

Sein zweiter Tip führt uns in die Fantasy-Welt, in der Rings of Medusa spielt.

Die Ringe der Medusa

Man muß eigentlich nur im Landschaftsmodus den Namen 'desoxyribonukleinsaeure' eintippen und schon gelangt man in eine Cheat-Seite, die man mit Escape wieder verlassen kann. Die Biologen und Chemiker unter uns sind sich allerdings einig, daß es die Abkürzung DNS oder englisch DNA auch getan hätte. Apropos englisch, beim Eingeben des Namens der Säure ist auf die amerikanische Tastaturbelegung zu achten.

Ein weiterer Tip von ihm führt uns wieder in die Gegenwart, an die Strände der Welt.

Beach Volley

Gibt man hier während des Spiels 'DADDYBRACEY' ein und drückt anschließend die

Taste F1, kommt man einen Level weiter.

Rock'n'Roll ist sowohl als Tanz auch als Computerspiel immer noch sehr beliebt, und so bringt sein letzter Tip einige Hilfen hierzu.

Rock'n'Roll

Als eigenen Namen sollte man 'MAGIC MAP' eingeben, worauf man die Karte in der höchstmöglichen Auflösung bekommt.

Gibt man als eigenen Namen 'COUNTRY' ein, gelangt man in ein Menü, in dem man sich alle Musiktitel anhören kann.

Wenn man aber als eigenen Namen Rainbow Arts angibt, so gelangt man in ein Eingabefenster mit zwölf Nullen. Hier kann man die Nummer des Levels angeben, in dem man gerne starten möchte. Und zwar folgendermaßen: Man gibt die Nummer des Levels ein, und zwar zweistellig, gefolgt von zwei X. Nun kommt eine vierstellige Zahl, deren Quersumme die Zahl des Levels ergeben sollte, in den man will, wieder gefolgt von zwei X. Die letzten beiden Nullen ersetzt man durch die Ziffern des gewünschten Levels in umgedrehter Reihenfolge. Wem das zu schwer war, hier ein Beispiel für den Level acht:

08xx2312xx80

Auch von Simon Smoley aus Spittal in Österreich bekamen wir Tips, von den wir ein paar abdrucken möchten.

Populous

Wer bei Populous in höheren Leveln nicht mehr weiter kommt, kann mit den folgenden Passwörterm aufatmen:

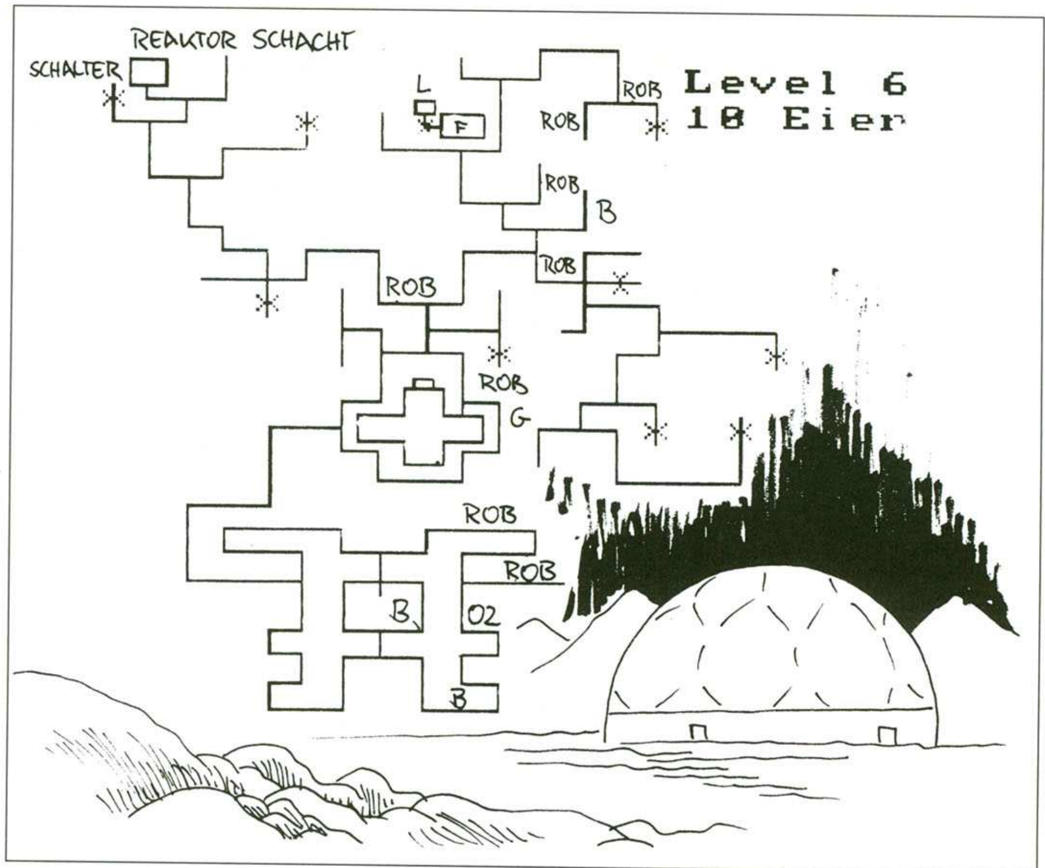
CALEOLD
EOAMPMET
BADMEILL
SHADWILDON

Des weiteren hat er noch einige ergänzende Tips zum Spiel Hybris, aus der Ausgabe 4/90.

Hybris, die Zweite

Nach der Eingabe von COMMANDER in der Score-Liste sind auch die Funktionstasten F1-F8 belegt. Drückt man während des Spiels auf eine dieser Tasten, bekommt man eine viel bessere Bewaffnung. Außerdem verfügt man über unendlich viele Megabomben.

Markus Schmid aus Waldenbuch sendet uns einen Tip zu Batman, The Movie.



Infestation-Karte, Level 6

Fledermausmann

Im Titelbild, während Batman und Joker abgebildet sind, gibt man einfach J,A,M ein. Den letzten Buchstaben, also das 'M' muß man so lange wiederholen, bis sich das Bild auf den Kopf dreht. Wenn man jetzt das Spiel startet, kann man mit Hilfe der Taste F10 jeweils einen Level überspringen.

Infestation

Wir wollen uns, auch wenn die Osterzeit schon vorbei ist, auf die Suche nach Eiern bei Infestation begeben. Zunächst einmal benutzt man draußen am besten die MMU, damit die zahlreichen Roboter nicht an uns rankommen. Wir fliegen zu den Koordinaten 90,30, wo sich ein Terminal befindet, in das wir unseren Namen, also KAL SOLAR, eingeben. Nach Eingabe des letzten Zeichens wandelt sich das "deaktiviert" in "aktiviert" und wir können den Transporter bei 50,50 in Richtung der Pfeile betreten. Die Dropzone, falls das Schiff dort etwas abgeworfen hat, befindet sich bei 45,10. Nachdem wir den ersten Level, abgesehen von der dunklen Zone, von Eiern gereinigt haben, besorgen wir uns die Lift-Berechtigungskarte aus dem Luftschacht und gehen in den Lift. Um in den Luftschacht hineinzugelangen, brauchen wir aber noch den Schraubendreher. Hier fahren wir nach ganz unten, in die sechste Ebene. Dort muß zunächst der Code für die Energiesperre am Terminal geknackt werden. Dazu müssen wir die Rechtecke so verschieben, daß das große Rechteck dem kleinen rechts unten in der Ecke entspricht. Nachdem dies geschehen ist, holen wir uns den Schlüssel für die Quartiere und begeben uns in den Luftschacht, um im Reaktorraum den Schalter umzulegen, der die Kühlung einschaltet. Täten wir dies nicht, würde der Kern nach genau 60 Minuten unseres Aufenthalts explodieren. Das ist auch die Zeit, in der die Aliens ganz aus ihren Eiern herauskommen. Aufbrechen tun sie aber schon nach dreißig Minuten. Nachdem auch dies geschehen ist, brauchen wir eigentlich nur noch die Computer-Link-Karte aus dem zweiten Level, die zusammen mit dem Kompaß die Orientierung sehr erleichtert und die Infrarotbrille aus Level 3, um die Suche nach den Eiern erfolgreich zu beenden. Während des gesamten Spiels sollte der Helm geschlossen sein, damit man vor unliebsamen Überraschungen geschützt ist. An Sauerstoffmangel stirbt man glücklicherweise nicht.

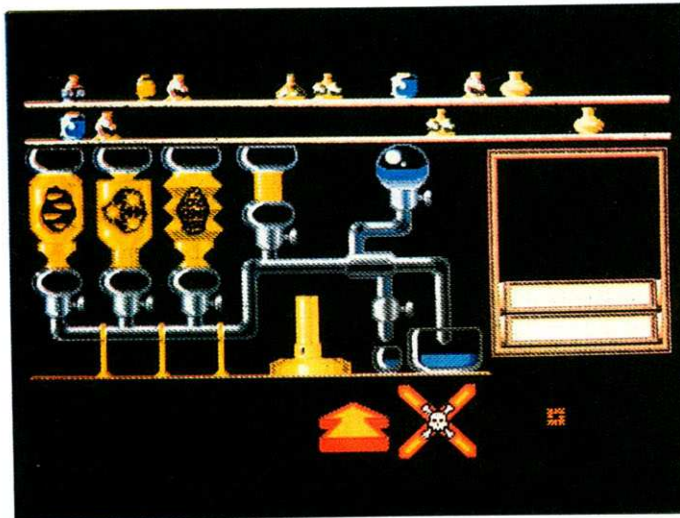
Legende zu Infestation

Folgende Abkürzungen treten bei den Karten zu Infestation immer wieder auf:

+,*:	Ei
A:	Quartiersschlüssel
C:	Computerkarte
F:	Essen
G:	Gas
L:	Lift
O2:	Sauerstoff
Rob:	Roboter
S:	Boulder (Vorsicht, stark radioaktiv!)
SD:	Schraubendreher
T:	Terminal
Telep:	Teleporter
W:	Wächter

So, das war es wieder einmal. Viel Spaß beim Spiel bis nächsten Monat wünscht das AMIGA-DOS-Team.

(Robert Marz/hs)



Wirtschaft in einer Fantasy-Welt – auch hier gelten gewisse Gesetze, die es zu nutzen gilt

Dragons Breath

Wirtschaftssimulation und Strategiespiel einmal anders: In Dragons Breath muß mit minimalen Resourcen eine Flotte schlagkräftiger Drachen herangezüchtet werden.

Schauplatz des Spiels ist das Land Anrea, eine Fantasy-Welt jenseits von Zeit und Raum. Dort soll im Zwergenberg das Geheimnis der Unsterblichkeit zu finden sein. Doch nur wer im Besitz der drei Teile eines Talismans ist, vermag dieses Geheimnis zu ergründen – allerdings sind diese Teile im Lande weit verstreut und müssen erst gefunden werden. Also machen sich drei (aus heutiger Sicht eher zwielichtige) Typen auf, um die Teile zu finden. Dies können sie jedoch nicht persönlich tun, da die Teile nur von ausgebildeten Drachen gefunden werden können. Also wird der Spieler zum Drachenmeister, der sich um die Aufzucht und den Einsatz der Drachen kümmern muß.

Die Karten stehen zu Anfang gar nicht so gut, da nur ein Drache vorhanden ist. Andere Drachen müssen erst in einer Brutkammer ausgebrütet werden, was nicht nur Zeit, sondern auch Geld für die Heizung kostet. Das wiederum fällt auch nicht vom Himmel, sondern muß in Form von Steuern eingetrieben werden. Dazu muß man jedoch erst mal einige der umliegenden Dörfer erobern, was nur mit Drachen möglich ist, deren Aufzucht wiederum... Es ist also gar nicht so einfach, unsterblich zu werden. Wenn man nicht genau mit seinen Resourcen haushaltet, steht man schnell vor dem Ruin – besitzt man weder Geld noch Drachen, ist das Spiel vorbei.

Um sich das Leben etwas zu erleichtern, kann man auch auf Magie zurückgreifen. In der eigenen Hexenküche können verschiedene Zutaten gemischt und appliziert werden. Die Ziele der Zaubereien sind breit gefächert. Man kann Dörfer vor einer Eroberung schwächen oder stärken, um ihre Steuererträge zu erhöhen, man kann jedoch auch seine Drachen dopen oder das Brüten beschleunigen. Doch auch das Zaubern hat seine Nachteile: Zum einen kosten die bei fahrenden Händlern erhältlichen Zutaten Geld, zum anderen sind einige Experimente nötig. Es sind nämlich nur die generellen Tendenzen der Komponenten sowie einige wenige 'sichere' Kombinationen bekannt, andere Mixturen können unerwünschte Nebeneffekte haben oder gar die Zauberküche zerstören.

Bis zu drei Spieler können an der Hatz nach dem Talisman teilnehmen. Stehen diese nicht zur Verfügung, so übernimmt der Computer die übrigen Figuren. Die Bedienung erfolgt komplett mit der Maus und beschränkt sich überwiegend auf das Anklicken von Icons oder Objekten. Grafik und Sound sind bei Dragons Breath sehr gut gelungen, können über eine kleine Schwäche des Spiels jedoch nicht hinwegtäuschen: einer dauernden Gratwanderung zwischen Begeisterung und Langeweile. Zu Anfang ist die Aufgabe sicher reizvoll: Es wird viel für Auge und Ohr geboten und die Verwaltung der Mittel erfordert viel Umsicht. Nach 10 oder 20 Stunden kommt jedoch der Punkt, an dem sich nichts neues mehr tut.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Dragons Breath
Hersteller: Palace
Vertrieb: Fachhandel
Preis: 99,95 DM

Positiv:
- Grafik und Sound gut
- komplexe Aufgabe
- unterstützt zwei Laufwerke
- Programm und Anleitung in deutsch

Negativ:
- hoher Preis
- kann langweilig werden





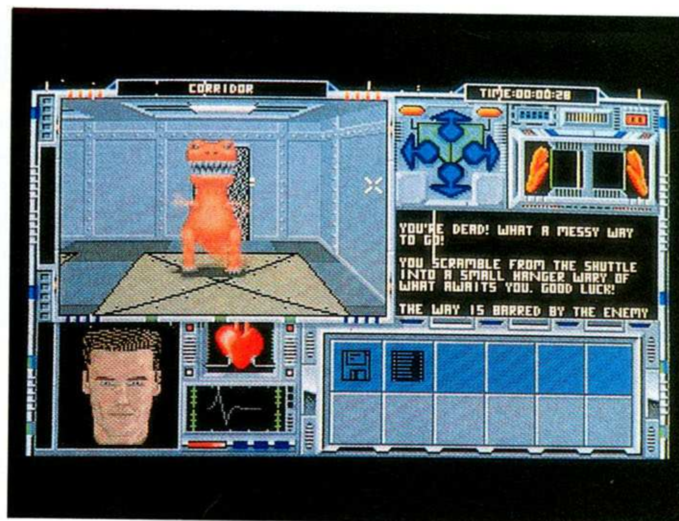
Drachen sind das A und O dieses Spieles, ohne sie ist ein Sieg unmöglich



Unsterblichkeit ist der Preis, um den es in Dragons Breath geht



Die Abenteuer des Wilhelm Tell – jetzt als Computerspiel



Sieht so ein Empfangskomitee aus?

Crossbow

Spaß, Spannung und Abenteuer mit Wilhelm Tell. Der bekannte Schweizer Nationalheld hat es, nach seinem englischen Kollegen Robin Hood, nun endlich auch geschafft, ein Computerspielheld zu werden.

In einer Mischung aus Adventure und Actionspiel übernimmt der Spieler die Rolle des bekannten Eidgenossen, der sich mit dem größenwahnsinnigen Herzog Gessler angelegt hat. Neben Tell wird das Land von vielen anderen Figuren bevölkert. Da gibt es Mönche, Ritter, arme Leute und viele wilde Tiere. Außer seiner Armbrust, die Tell ohnehin schon bei sich hat, kann er im Verlauf des Abenteuers noch viele andere Waffen und weitere nützliche Gegenstände finden. Seine Gesundheit wird durch eine Schwertklinge symbolisiert. Je kürzer die Klinge ist, desto schlechter geht es Wilhelm. Lebensmittel wirken dem Sinken der Energiereserven entgegen.

Muß Wilhelm kämpfen, gilt es, die richtige Waffe für den entsprechenden Gegner zu finden.

Das Szenario stellt sich als Querschnitt dar, durch den Tell wandert. Die Hütten der armen Bauern sind dabei genauso Schauplatz wie Kerker, Wiesen und Wälder.

Es ist ziemlich leicht, in der Landschaft dieses Spiels die Orientierung zu verlieren, da viele Schauplätze sehr ähnlich aussehen. Die Steuerung und die Auswahl einzelner Gegenstände aus dem Inhaltsverzeichniss geschieht komplett über Icon. Leider erscheinen die meisten Gegner recht plötzlich und bewegen sich recht schnell, so daß es schwierig ist, rechtzeitig die richtigen Icons zu betätigen.

Alles in allem ist Crossbow ein spannendes Action-Adventure, das zwar solide gemacht ist, aber auch echte Schwächen hat.

(hs)

AMIGA DOS Blitzlicht

Name: Crossbow
Hersteller: Screen 7
Quelle: Fachhandel
Preis: stand bei Redaktionsschluß nicht fest

Positiv:
- passable Grafik
- auch Kopfarbeit erforderlich

Negativ:
- Steuerung gewöhnungsbedürftig



Federation Quest One – BSS Jane Seymour

Seltsame Kreaturen läßt Mutter Natur (oder Vater Gentechniker) entstehen, wenn man ihr (oder ihm) nur Zeit genug läßt. So geschehen auf dem verschollenen Sternenschiff 'Jane Seymour'.

Vor zehn Jahren brach die 'Jane Seymour' zu einer Forschungsreise durch die unbekannten Gebiete der Galaxis auf, um fremde Lebensformen kennenzulernen und zu studieren. Ein einziger falsch berechneter Hypersprung genügte allerdings, um die Mission scheitern zu lassen: Die an Bord befindlichen Besatzungsmitglieder wurden harter Strahlung ausgesetzt, erkrankten und vergaßen ihre ursprüngliche Aufgabe; die bereits eingesammelten fremden Lebensformen kamen frei und verunsichern jetzt die Umgebung.

Da stehen Sie nun frisch angekommen im Hangar und sehen alles in schönster Unordnung. Gegenstände liegen herum, die Sie am besten mitnehmen, wenn sie Ihnen nützlich erscheinen. Ihre augenblicklich besten Freunde sind eine Handvoll Roboter, die Sie mit speziellen Modulen füttern müssen, um sie richtig ans Arbeiten zu bekommen.

Die Aufgabe, das Schiff wieder flottzumachen, die Besatzung wieder zu aktivieren und die Aliens einzusperren, ist nicht leicht.

Das 'Jane-Seymour'-Abenteuer verspricht den Freunden von Action-Adventures deshalb manch spannende Stunde, auch wenn einige Aktionen etwas umständlich erscheinen. Eine spannende Sache ist das Spiel aber allemal.

(jb)

AMIGA DOS Blitzlicht

Name: Federation Quest One – BSS Jane Seymour
Hersteller: Gremlin
Quelle: Fachhandel
Preis: stand bei Redaktionsschluß noch nicht fest

Positiv
- spannende Handlung
- viele Räume mit vielen Gegenständen
- Kombinations- und Reaktionsvermögen gefordert

Negativ:
- Gegenstände etwas umständlich unterzubringen und zu bedienen
- Sound nicht ausgereizt





Der Kämpfer gegen die Langeweile hat seinen Kampf verloren. Eine sehr müde Ballerei



Im heimatlichen Dojo wird für die Meisterschaft geübt

Scramble Spirits

Das Verhältnis zwischen Gut und Böse hat sich mal wieder ganz entschieden in Richtung Böse verschoben. Um diesen Zustand zu ändern, besteigen Sie allein oder mit einem Freund ihre Düsenjäger und treten gegen die Geschwader der Bösen an.

Mit Ihrem Düsenjäger fliegen Sie über die vertikal scrollende Landschaft und ballern, was das Zeug hält. Ihr Jäger ist mit Maschinengewehren für die Luft- und mit Bomben für die Bodenziele ausgerüstet. Beide Waffensysteme werden über den Feuerknopf des Joysticks gleichzeitig ausgelöst. Als Extrawaffen fliegen von Zeit zu Zeit kleine Flugzeuge über den Bildschirm, die sich, wenn man sie berührt, an die Seite des eigenen Fliegers hängen und kräftig mitschießen. Durch einen langen Druck auf den Feuerknopf verwandeln sie sich in Bomben, die alle auf dem Monitor vorhandenen Gegner vernichten.

Zu Beginn des Spiels stehen fünf Credits zur Verfügung, jedes mit fünf Leben. Spielt man allein, kann man fünfmal die Continue-Option nutzen. Fliegt ein Freund mit und hilft beim Aufräumen unter den Aliens, so zehrt er ebenfalls von diesen Credits. Ab und zu gibt es noch eine Bonussequenz, bei der das eigene Flugzeug und die Gegner vergrößert dargestellt werden und ohne Gegenwehr darauf warten, abgeschossen zu werden.

Scramble Spirits ist ein sehr laues Spiel, das zwischen Extremen schwankt: Entweder die Gegner kommen in solchen Massen, daß man kaum eine Chance hat, oder man befindet sich in einer langen Phase, während der überhaupt nichts passiert.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Scramble Spirits
Hersteller: Grandslam
Quelle: Fachhandel
Preis: 64,95 DM

Positiv:
- passabler Sound
- Zwei-Spieler-Modus
- Continue-Option

Negativ:
- schwache Grafik
- laues Spielgeschehen
- lange Leerflugphasen



Budokan

Kampfsportsimulationen gibt es wie Sand am Meer. Aber nur wenige bemühen sich, die dargestellte Sportart auch wirklich realistisch zu zeigen.

Bei Budokan ist das anders, denn der Programmierer selbst ist Träger des 3. Dan, also des dritten schwarzen Gürtels. Der Schüler betritt zunächst einmal die Dojo, die Übungshallen der Schule. Jede der Techniken wird in einer gesonderten Halle trainiert. Hier kann der Budoka (Schüler) zwischen Karate, Nunchaku, Kendo und Bo auswählen.

Karate dürfte gemeinhin bekannt sein, auch unter einem Nunchaku können sich die meisten wohl noch etwas vorstellen. Schwieriger wird es dann schon bei Bo und Kendo: Bo ist der Kampf mit einem Holzstab, Kendo der Kampf mit der Übungswaffe der Samurai, einem Bambusschwert. In den Hallen kann der Kämpfer entweder allein trainieren oder sich im 'Kumite' mit einem Lehrer versuchen. Nach Beendigung eines solchen Kumites gibt der Lehrmeister wirklich brauchbare Tips. Fühlt man sich in den Kampftechniken genügend geschult, kann man auf der Matte mit einem gleichgesinnten Schüler kämpfen. Hierbei sind auch Kombinationen wie Karate gegen Kendo möglich. Wer lange und hart genug trainiert hat, kann vielleicht im Budokan-Gebäude gegen die besten Kämpfer der Welt bestehen. Budokan ist ein Kampfspiel, das nicht nur den Aspekt Kampf in den Mittelpunkt des Spielgeschehens rückt.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Budokan
Hersteller: Elektronik Arts
Quelle: Fachhandel
Preis: ca. 89 DM

Positiv:
- komplexe Simulation
- viele verschiedene Sportarten
- kein Kopierschutz

Negativ:
- nicht auf Harddisk installierbar
- lange Ladezeiten





Bestreiten Sie mit Ihrem Rotor beinahe unmögliche Himmelfahrtskommandos

Rotor

In nicht allzu ferner Zukunft müssen alle bis auf eine kleine Oberschicht unter sklavenartigen Bedingungen arbeiten. Die jungen Leute haben jedoch eine Alternative: Sie können als Rotorpilot zu Ehre, Ruhm und Geld gelangen.

Die Rotoren sind Flugzeuge, die speziell für den Einsatz hinter den feindlichen Linien gebaut wurden. Sie verfügen über keinerlei aerodynamische Eigenschaften und über nur eine Düse am Heck, mit der es gilt, gegen die Schwerkraft zu bestehen. Sie sollen den Feind im eigenen Gebiet schwächen und Materialien wie Treibstoff und vor allem Sonnenkristalle stehlen.

Vor den Einsatz haben die Militärs den Übungsflug gesetzt und so müssen Sie ihre Fähigkeiten zunächst im Simulator unter Beweis stellen. Ihre Erfolgsbilanz wird in einem Mikro-Taschencomputer festgehalten, der anhand der gesammelten Prestigepunkte, die man für Abschüsse etc. erhält, entscheidet, wann Sie für eine höhere Missionsstufe zugelassen werden. Haben Sie genügend Punkte, bekommen Sie ein Paßwort und die nächsten drei Missionen werden Ihnen zugewiesen.

Sie können Ihren Rotor erweitern, indem Sie Perlen sammeln, die Icons auf einem Transformatorbalken wandern lassen. Wird der Transformator aktiviert, erhalten Sie das entsprechende Extra. Wird Ihr Rotor zerstört, gehen die Position des Balkens sowie die Prestigepunkte der aktuellen Mission verloren. Die Ausrüstung des Schiffes wird allerdings auf den nächsten Rotor übertragen.

Rotor ist ein Spiel für lange Abende, das den Spieler vor den Bildschirm zu bannen vermag.

(Robert Marz/hs)

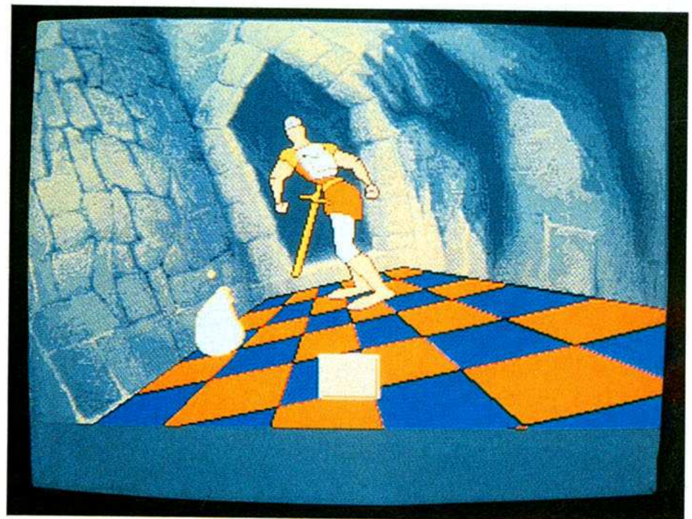
AMIGA DOS Blitzlicht

Name: Rotor
Hersteller: Arcana
Quelle: Fachhandel
Preis: 69,95 DM

Positiv:
- Passwortsystem
- gute Grafik

Negativ:
- fast kein Sound





In das Schloß hineinzukommen, um die Prinzessin zu retten, war schon nicht einfach. Aber schaffen Sie es auch, wieder aus dem Schloß zu entkommen?

Escape from Singe's Castle

Dragon's Lair geht weiter: Der böse Drache Singe hat schon wieder die schöne Prinzessin Daphne gefangengenommen und in die innersten Gemächer seines Schlosses gebracht. Wird es Ihnen gelingen, die liebliche Daphne den Klauen Singes zu entreißen?

Dragon's Lair, ein Actionspiel, das schon seit längerem auf dem Amiga zu überzeugen vermochte, hat einen Nachfolger bekommen. Escape from Singe's Castle ist die neueste Episode aus den Abenteuern von Dirk dem Kühnen. Wieder einmal muß er gefährliche Abenteuer meistern, bevor er im Endkampf mit dem heimtückischen Drachen Singe die Prinzessin Daphne erretten kann.

Der zweite Teil des fast zeichentrickartigen Spiels Dragon's Lair hat einige neue Features hinzubekommen, die beim ersten Teil leider nicht zu finden waren. So ist es nun möglich, das Spiel auf eine Festplatte zu installieren, eine Linkoption befähigt den Benutzer sogar, Dragon's Lair einzubinden und somit auch auf der Festplatte zu verewigen. Ferner werden nun auch diejenigen, die nur über 512 kByte verfügen in den Genuß kommen, den Helden durch die dunklen Gemäuer zu begleiten. Ein Trainer-Modus zeigt dem Spieler die jeweils zu tätigende Joystickbewegung an. Standen bei Dragon's Lair nur drei Leben zur Verfügung, so kann der Anwender nun bis zu fünf Heldenleben fordern.

(br)

AMIGA DOS Blitzlicht

Name: Escape from Singe's Castle
Hersteller: Visionary Ltd.
Vertrieb: Fachhandel
Preis: ca. 140 DM

Positiv:
- gute Grafiken
- auf Festplatte installierbar
- Dragon's Lair kann eingebunden werden
- auch mit 512 kByte lauffähig

Negativ:
- hoher Preis





Turrican – Action vom Feinsten!



In diesem Raumschiff geht der Bär, äh der Alien ab. Legen Sie sich schon mal einen Ersatz-Ballerdaumen zurecht

Turrican

Was erzeugt Herzkammerflimmern ohne Strom, Schweißausbrüche ohne Schwerstarbeit, Angstzustände ohne Samstagabend-Krimi – kurzum Spannung bis zum "Gehtnichtmehr"? Ganz einfach: Turrican.

Die Story ist so alt wie die Computerspiele selbst: Guter Held rettet schöne Welt vor garstigem Ungeziefer. So auch hier: Der Held von Turrican muß sich gegen eine ganze Menge von Unholden zur Wehr setzen, die eine fast unüberschaubare Station beherrschen. Mit bloßen Fäusten ist da nichts zu machen, also muß schweres (Laser-)Geschütz her. "Mit der Waffe in der Hand kommt man hier durch jede Wand" – eine Devise, die hier ganz nützlich sein kann, denn nicht jede Wand ist undurchdringlich. Überall kann der Held des Spiels Edelsteine finden, die ihm Ruhm und Ehre oder auch verlorene Lebensenergie zurückbringen. Ab und zu findet man auch einen besonderen Stein, der nach der Zerstrahlung Bonuspunkte en masse abgibt. Noch besser sind die unsichtbaren Quader, die erst nach Beschuß sichtbar werden, dann aber ein ganzes Kontingent an Zusatzpunkten, wie verbesserte Waffen, Unverwundbarkeit und ähnliches freigeben.

Bei Spielen, deren Thematik schon Staub ansetzt, ist man ja irgendwie skeptisch, bei Turrican gibt es dazu jedoch keinen Grund. Das Spiel ist nicht nur abwechslungsreich, sondern auch schnell, grafisch gelungen und mit gutem Sound unterlegt. Kurzum: ein Muß für jeden Action-Liebhaber.

(jb)

AMIGA DOS Blitzlicht

Name: Turrican
Hersteller: Rainbow Arts
Quelle: Fachhandel
Preis: Steht noch nicht fest

Positiv
- sehr gute Grafik
- guter Sound
- fließende Animation
- nicht zu schwer

Negativ
- nichts feststellbar



Leavin' Teramis

Die Teramis ist von Aliens besetzt worden und Sie müssen nun natürlich versuchen, die Aliens dahin zu schicken, wo sie herkamen, und das geht in diesem Falle nur mit Waffengewalt.

Die Teramis ist ein sehr großes Raumschiff, das sich in viele Sektionen unterteilt. Sie fangen mit ihrer Säuberungsaktion ganz hinten in der Hyperdrive-Sektion an, darauf folgen dann die Energie-, die Pflanzen- und die Freizeit-Sektion und so geht es weiter. Während Sie also durch das Schiff rennen und kräftig ballern, sehen Sie sich und Ihre nähere Umgebung aus der Draufsicht. An manchen Stellen, die durch Pfeile gekennzeichnet sind, können Sie eine Plattform betreten, die sich außen am Schiff befindet. Auf der Plattform befindet sich eine Falltür, die sich öffnet und einen mehr oder weniger nützlichen Gegenstand freigibt. Dies wären zum Beispiel zusätzliche Zeit oder eine bessere Waffe. Auf manchen von diesen Seitengängen haben sich allerdings bereits Aliens eingenistet, die Sie vorher erst noch erledigen müssen. Doch Vorsicht, manchmal haben sie sich auch unter der Plattform versteckt und greifen erst an, wenn man sich in Reichweite befindet. Auch im Schiff selbst liegen Icons herum, die, wenn man sie aufnimmt, Extrawaffen, neue Leben oder Punkte bedeuten. Und natürlich hat jede Sektion ihren Oberalien, der erst besiegt werden muß, bevor es in den nächsten Level geht. Baller-Freunden kann man dieses Spiel nur wärmstens empfehlen, denn hier geht der Bär ab.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Leavin' Teramis
Hersteller: Thalion
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- viel Action
- viele Extras
- kernige Aliens

Negativ:
- lange Ladezeiten





Preisfrage: Wo hat Coloris die Spielidee geklaut?



Viel feine Grafik in einem erstaunlich guten Actionspiel

Coloris

Tetris hat es nicht nur Tausenden von Computerspielern angetan, für die Gilde der Spieleprogrammierer stellt es ein beliebtes Vorbild dar, dem oft und gerne nachgeeifert wird.

Auf den ersten Blick unterscheidet sich Coloris im grundsätzlichen Spielaufbau nur unwesentlich von Tetris. Auch hier fallen Blöcke in einen Auffangbehälter. Die fallenden Blöcke gleichen sich in der Form, längliche Rechtecke, die in drei Segmente unterteilt sind. Jedes dieser drei Segmente kann eine unterschiedliche Farbe haben. Aufgabe des Spielers ist es, die Blöcke so zu plazieren, daß die verschiedenfarbigen Segmente in horizontaler oder vertikaler Ausrichtung zu Gruppen angeordnet werden. Sobald drei gleichfarbige Segmente in einer Reihe nebeneinander sitzen, werden Sie aus dem Haufen entfernt und etwa darüberliegende Blöcke rutschen nach. Der Spieler kann die herabfallenden Steine nach links und rechts lenken oder durch Drücken des Feuerknopfes die farbigen Segmente der Blöcke rotieren lassen. Gelingt es dem Spieler, mehr als drei Steine in einer Reihe zu gruppieren, gibt es, je nach Menge der Steine, einen Bonus in Form von verschwindenden Blöcken. Türmen sich die herabgefallenenen Steine bis über eine besondere Markierung, ist das Spiel beendet.

Auch wenn das Konzept, das Tetris zugrunde liegt, recht strapazierfähig ist, ist dieser neue Epigone nur absolut Tetris-Süchtigen zu empfehlen oder jenen, die das Spiel noch nicht besitzen. Ansonsten ist das Thema, trotz sauberer Programmierung, zweckmäßiger Grafik und passabler Musik (düstere 'Grufti-Rock'-Synthesizer-Klanggebilde), nur wenig variiert und damit dem Original nur allzu ähnlich.

(hs)

AMIGA DOS Blitzlicht

Name: Coloris
Hersteller: Signum Victoriae
Quelle: Fachhandel
Preis: 49,95 DM

Positiv:
- originelle Musik

Negativ:
- ziemlich betagtes Spielkonzept



Black Tiger

Drei fürchterliche Drachen halten die Stadt in Angst. Keiner wagt es, sich gegen die Bestien und ihren feurigen Atem aufzulehnen, bis ein mächtiger Krieger die Szene betritt: der schwarze Tiger.

Das Land hat sich verwandelt. Unholde schleichen durch die Ruinen und die wenigen ehrlichen Menschen wurden durch die Drachen in Stein verwandelt. Aber ein mächtiger Krieger wie der schwarze Tiger ist nicht wehrlos. Er schwingt seinen Morgenstern, schleudert Angreifern Messer entgegen und sammelt deren Schätze, die in Form von Münzen zurückbleiben, ein. Mitunter kommt der Tiger auch an einem der armen Versteinerten vorbei, die auf Rettung warten. Von ihnen kann er Münzen oder hilfreiche Tips bekommen. Einige besitzen sogar ein ganzes Warensortiment, aus dem sich Tiger, sofern sein Geld reicht, Rüstungen und bessere Waffen besorgen kann. Black Tiger ist ein Jump'n'run-Spiel mit viel Action. Die Grafik ist brillant gestaltet, auch wenn sie nicht die ganze Grafikpower des Amiga ausnutzt. Black Tiger überzeugt aber vor allem durch sein fesselndes Spielgeschehen. Jedes Level ist ein großes Gelände, in dem Tiger herumlaufen und klettern kann und das er nach Schätzen, Monstern und versteinerten Menschen durchsuchen muß. Am Ende jedes Levels erwartet den Tiger ein besonders böses Supermonster, das besiegt werden muß.

Sicher gibt es auch an Black Tiger Mängel, so ist das Scrolling doch recht ruckelig ausgefallen, trotzdem ist das Programm ein spannendes Actionspiel, das sich ob seiner Qualitäten wohltuend von der grauen Masse ähnlicher Produkte abhebt.

(hs)

AMIGA DOS Blitzlicht

Name: Black Tiger
Hersteller: U.S. Gold
Quelle: Fachhandel
Preis: 84,95 DM

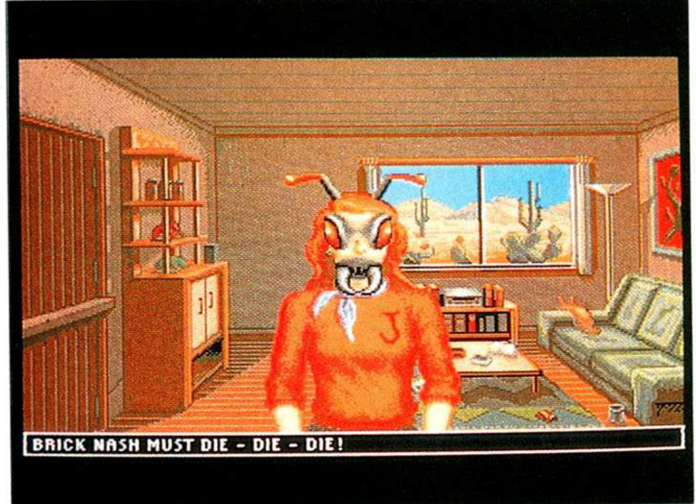
Positiv:
- gute Grafik
- spannender Spielverlauf

Negativ:
- Scrolling ruckelt





Labyrinthfreunde werden sich freuen, in Puffy's Saga gibt es mehr als genug Irrgärten zu erforschen



Seitdem eine Ladung Plutonium-Zündkapseln gestohlen wurde, gehen mit den Bewohnern von Lizard Breath seltsame Veränderungen vor

Puffy's Saga

Wie Puffy und Puffyn in das geheimnisvolle Labyrinth hineingerieten, wissen sie selbst nicht mehr so genau. Auf jeden Fall stecken sie nun inmitten eines ausgedehnten Irrgartens, angefüllt mit Schätzen und Untieren.

Gespenster, Drachen, garstige Monsteraugen und Säuretropfen treiben sich in den Mauern herum und warten nur darauf, Puffy oder Puffyn Lebensenergie abzusaugen. Doch die beiden, die entweder allein oder gemeinsam auf Erkundungstour gehen können, verstehen sich zu wehren und spucken im wahrsten Sinne des Wortes auf die Gefahr. Der gefahrvolle Weg durch die Gewölbe läßt sich nur mit Köpfchen und Mut meistern. Hier öffnet eine Bodenplatte Geheimtüren, dort vermag ein Schlüssel zu helfen und manchmal lassen sich sogar die Mauern einfach 'niederspucken'. Neben allerlei Monstergetier wird das Leben von Puffy und Puffyn auch durch eine ganze Reihe brandgefährlicher Baulichkeiten bedroht. Da gibt es fiese Pieken, die aus dem Boden wachsen und Generatoren, die regelrechte Blitzgewitter abfeuern. Neue Energie können die beiden Helden durch das Einsammeln von Schinken erhalten, die an vielen Stellen im Labyrinth versteckt sind.

Vor einiger Zeit gab es auf dem Amiga ein Spiel, das nannte sich Garrison. In der Tat ähnelt Puffy's Saga diesem Spiel verblüffend. Jedoch zeigt sich schon nach wenigen Spielen, daß hier mehr auf Denkarbeit und Strategie Wert gelegt wird. Viele Levels sehen auf den ersten Blick so aus, als sei es unmöglich, sie zu durchwandern, und erst nach einigen Versuchen kristallisiert sich eine Strategie heraus, die richtig angewandt, zum Ziel führt.

Ein bunt gestalteter Spielspaß der neben einigen Schwächen auch seine starken Seiten hat.

(hs)

AMIGA DOS Blitzlicht

Name: Puffy's Saga
Hersteller: UBI Soft
Quelle: Fachhandel
Preis: 89,– DM

Positiv:
- gute Grafik
- passables Spielgeschehen

Negativ:
- zu wenige Levels



ANTHEADS

Nachdem den Riesenameisen in 'It came from the desert' (siehe Bericht in Ausgabe 3'90) der Garaus gemacht wurde, scheint wohl in dem kleinen Wüstenstädtchen Lizard Breath Ruhe eingekehrt zu sein. Der Schein trügt jedoch, wie die Fortsetzung des Spieles beweist.

Eine Ladung Plutoniumkapseln, die in einem überirdischen Atombombentest eingesetzt werden sollen, verschwindet auf unerklärliche Weise. Nicht ahnend, daß die Riesenameisen wieder zurückgekehrt sind und nach dem lebenverleihenden Plutonium lechzen, macht sich Brick Nash, ein im Koreakrieg erprobter Ex-Kampfpilot auf die Suche danach. Prompt stößt er auch auf die mutierten Riesenameisen und versucht, die Bevölkerung von Lizard Breath zu warnen und die Ungetüme zu eliminieren. Hierbei spielt das Krankenhaus eine große Rolle, denn ein verrückter Arzt experimentiert mit seinen Patienten, die sich nach dem Hospitalaufenthalt in Ameisenmenschen verwandeln. Der zweite Teil von It came from the Desert steht dem ersten in nichts nach, obwohl sich nur wenig Neues ergeben hat.
Zum Installieren von Antheads benötigt man mindestens ein MByte RAM sowie den ersten Teil des Spieles.

(br)

AMIGA DOS Blitzlicht

Name: Antheads
Hersteller: Mirrorsoft
Vertrieb: Fachhandel
Preis: ca. 50 DM

Positiv:
- abwechslungsreich
- gute Grafiken, guter Sound
- Abspeichermöglichkeit

Negativ:
- umständliche Installation
- benötigt 'It came from the Desert' (Teil I)
- Ein MByte RAM erforderlich



Gesellschaftsspiele am Computer

'Das Magazin' und 'Das Haus'

Warum sollen Gesellschaftsspiele eigentlich immer ohne Computer stattfinden? Zwei neue Wirtschaftssimulationen mit den Namen 'Das Haus' und 'Das Magazin' sollen das Gegenteil beweisen: Der Computer als Spielfeld und die Spieler davor. Ob es gelingt?

Zukünftige Hausbesitzer können mit dem 'Haus' schon mal üben, wie man mit Mietern und Verbänden umspringt – oder diese mit dem Vermieter. Jeder Spieler erbt am Anfang ein Mietshaus, in dem eine Anzahl von Wohnungen schon vermietet sind. Jeder der neuen Hausbesitzer muß nun versuchen, aus seinem Haus noch mehr zu machen (mehr aus dem Haus und mehr Häuser, alles klar?). Ganz so einfach ist das jedoch nicht: Einfach mal die Miete erhöhen und kräftig absahnen – das läuft nicht in unserem Computerhaus. Setzt man die Miete zu hoch an, ziehen die Leute aus (wohin bleibt offen, angesichts der herrschenden Wohnungsnot können sie einem leid tun). Der Mietpreis muß also genau ausgepegelt werden, damit die Mieter zufrieden sind und der Profit trotzdem wächst.

Zwischendurch kann man an den Aktienmarkt gehen und hier sein Glück versuchen – wenn's klappt, klappt's!

Die Spielidee vom 'Haus' ist eigentlich ganz gut, die Umsetzung jedoch sehr mager. Außer ein paar Entscheidungen kann der Spieler nichts Weltbewegendes hervorbringen, so daß schnell Langeweile aufkommen kann. Die Aktionen werden höchstens einmal von 'Lottogewinnen', gefundenen Brieftaschen und ähnlichem unterbrochen, wobei aber kaum eine Hausbesitzer-Atmosphäre auftritt. Fazit: Wer Wirtschaftssimulationen sammelt, der wird wohl auch hier zugreifen.

Wie macht man eine Computerzeitschrift? Mit Geld und dem 'Magazin'

'Das Magazin' schlägt in die gleiche Kerbe, allerdings sind die Programmierer mit mehr Liebe an das Projekt gegangen. Das Programm zeigt, wie man eine Computerzeitschrift macht...

Nein, beschreiben wir es anders. Denn für die Autoren dieser Zeilen ist das, was als Simulation vorliegt, harte Wirklichkeit, also kann man mit dem nötigen Ernst an den Test gehen!

Nein, so auch nicht. Das Magazin ist nämlich keine bierernste Sache, sondern eine mit reichlich (manchmal schwarzem) Humor gespickte Vortäuschung richtiger Tatsachen.

Begonnen wird mit einem erklecklichen Sümmchen, das jedoch weder in Häusern, Schiffen, Frauen und anderen geldverschlingenden "Gegenständen" angelegt wird, sondern der Gründung eines Computer-Magazins dient.

Und so wird erst mal der Top-Redakteur mit Barem geködert sowie die ersten drei jungen Schreibkünstler eingestellt. So eine Erstausgabe muß gut ankommen, also muß ein Preisausschreiben her, bei dem außer Goldbarren alles gewonnen werden kann, angefangen vom T-Shirt bis hin zum Drucker.

Ein bißchen Public Relations wäre auch nicht schlecht, kostet zwar nicht wenig, ist aber der Sache dienlich. Auch eine Umfrage ("Wie hat Ihnen der Inhalt des Impressums gefallen?") kann ganz interessant sein, zwar auch nicht gerade billig, aber besser, man weiß früh genug, was den Leser freut, als wenn statt der nächsten Ausgabe die Aufgabe folgt.

Auch die Werbetrommel muß gerührt werden. Nicht von der Zeitschrift, sondern von Firmen, die ihre Produkte über die Zeitschrift durch Anzeigen an den Mann oder die Frau bringen wollen. Fremdwerbungen bringen Geld, deshalb heißt es, vorher einen vernünftigen Preis dafür festzusetzen. Farbanzeigen bringen noch mehr Geld, also auch hier gilt es, das Preis-Leistungsverhältnis zu beachten. Je mehr Fremdwerbung im Heft ist, desto mehr wird eingenommen, logisch, oder?

Und dann passiert's! Das erste Heft geht in Druck. Die Seiten sind gefüllt mit Tests, Berichten (ach ja, die Auslandskorrespondenten sind angeheuert, zwar für immense Summen, aber das Heft steht), Hardware und allem, was dazugehört. Jetzt beginnt das große Zittern. Wird es ein Erfolg, oder wickeln die Marktfrauen die Fische drin ein?

Kurze Zeit später – das erste Ergebnis ist da. Die Zeitschrift hat sich gut verkauft, die Einnahmen decken nicht nur die Kosten, sondern bringen sogar einen Gewinn. Nur unter den 15 besten Zeitschriften hat Ihr Magazin noch nicht Einzug gehalten. Und Sie haben nur

AMIGA DOS Blitzlicht

Name: Das Haus
Hersteller: Ariola
Preis: 69,95 DM
Quelle: Fachhandel

Positiv:
- mehrere Spieler
- humorige Texte

Negativ:
- zu wenig Auswahlmöglichkeiten
- zu wenig Aktionen und Zufälle

AMIGA DOS URTEIL: SCHLECHT

GRAFIK	SOUND	MOTIVATION
2	2	4



Bild 1. 'Das Haus' erben Sie und damit auch die Sorgen und Nöte eines Vermieters

AMIGA DOS Blitzlicht

Name: Das Magazin
Hersteller: Ariola
Preis: 69,95 DM
Quelle: Fachhandel

Positiv:
- mehrere Spieler
- realitätsnah und doch witzig
- mausgesteuert

Negativ:
- etwas umständliche Menüauswahl

AMIGA DOS URTEIL: OK

GRAFIK	SOUND	MOTIVATION
7	3	6



Bild 2. Falls Sie sich mal wieder über eine Computerzeitschrift ärgern, machen Sie es doch besser. 'Das Magazin' ist Ihre Wunschzeitschrift



Mit Hot Rod kann jeder Amiga-Benutzer einen heißen Reifen fahren

ein Jahr Zeit, dies zu realisieren, andernfalls ist es Essig mit der Zeitschrift und mit Ihrem Job.

Klappt alles ganz wunderbar, wenn nicht ausgerechnet jetzt der Top-Redakteur seinen Sportwagen mit einem Baum kreuzen und damit gleich richtig aufsteigen würde – gen Himmel.

Also wird nach Ersatz Ausschau gehalten – und er auch gefunden. Schnell wird Geld geboten, um den Kandidaten in spe abzuwerben. Bange Frage: Nimmt er an? Ja, er nimmt an, zwar für ein Heidengeld, aber die Artikel sind gesichert.

Interessant sind auch die Umfragen. Trotz aller Veränderungen fanden die "Leser" unser Probemagazin "Stuß" (rheinisch: Unsinn). Und so endete alles nach einem Spieljahr mit einem hämischen Kommentar.

Einen direkten Vergleich zwischen beiden Spielen zu ziehen, ist nicht möglich, da die Themen unterschiedlich sind. Nur zur Spielart sind Vergleiche möglich: Während Das Haus mehr vor sich hinplätschert und nur bei Liebhabern von Simulationen Begeisterungsstürme entfachen wird, bietet Das Magazin einiges mehr. Nicht nur Redakteuren aus dem wirklichen Leben wird hier die Möglichkeit geboten, das Ruder im Zeitschriftenmeer herumzureißen, sondern auch diejenigen, die vom Zeitungswesen gar nicht viel kennen, haben hier die Chance, zum 'Verleger des Jahres' gekürt zu werden – sofern einem nicht allzu Schlimmes widerfährt. (Wie sicher ist Ihr Top-Redakteur am Steuer?) Mit mehreren Spielern macht es sowieso Spaß, weil man außer dem Namen auch noch den Computer- und Themenbereich wählen darf.

Gesellschaftspiele – einmal mäßig und einmal gut

Vielleicht ist ein gemeinsamer Spielabend mit 'Mühle', 'Dame', 'Skat' oder 'Mensch ärgere dich nicht' so manchem Spieleliebhaber wesentlich angenehmer, weil es auch ohne Computer geht, aber die Computer-Games sind nun mal da.

Daß man in fröhlicher Runde auch mit Unterstützung des 'Bit-Monsters' einen lustigen Abend verbringen kann, darauf setzen die beiden Spiele. Während Das Haus zwar mit Humor, aber wenig Aktion aufwartet, gibt Das Magazin schon mehr her.

Beide Spiele erfordern Entscheidungen des Spielers, jedoch kommt man sich beim "Haus" etwas zu sehr als Randfigur vor, während Das Magazin den Spieler mehr ins Geschehen einbindet.

Beide Spiele sind im Konzept ähnlich, wobei Das Magazin noch besser abschneidet.

Zwar sind beide Spiele keine Megahits, doch kann man zumindest Das Magazin empfehlen; für lange Abende ist es sicherlich willkommen.

(jb)

HOT ROD

Autorennen, gleichermaßen umstrittene Sportart und beliebtes Computerspielthema, allein, die Idee ist nicht gerade neu, und wer hier noch Lorbeeren erringen möchte, muß mit Qualität aufwarten.

Tatsächlich ist das Thema nicht neu und es gibt bereits mehrere Spiele, die sich diesem Sport ausgiebig widmen.

Hot Rod stellt die Rennstrecken aus der Vogelperspektive dar. Die einzelnen Kurse sind größer als ein Bildschirm und folgen dem führenden Fahrzeug durch Scrolling des Spielfeldes. Mitunter kann es passieren, daß ein Wagen zu langsam fährt und aus dem Bild scrollt. Dann wird er sozusagen 'herantele-portiert', so daß er wieder im sichtbaren Bereich der Rennstrecke ist. Da Hot Rod mit bis zu vier Teilnehmern gleichzeitig gespielt werden kann, ist diese Einrichtung bitter nötig. Ohne sie bliebe mancher Spieler im wahrsten Sinne des Wortes 'auf der Strecke'. Nach jedem Rennen gibt es für gute Plazierung Punkte und Preisgeld, das sich ausgezeichnet für den Erwerb zusätzlicher Ausrüstung eignet.

Vier Spieler gleichzeitig, da kommt Freude auf, denn ein solcher Wettstreit auf den vielen verschiedenen Motodromen von Hot Rod ist ein fesselndes Spielerlebnis. Insbesondere der mögliche Wettbewerb hebt dieses Spiel über vergleichbare Produktionen hinaus. Ansonsten wurde Hot Rod routiniert programmiert, ist mit zweckmäßigen Grafiken versehen und beinhaltet auch einiges an Musik und Sound-Effekten. Ein überdurchschnittliches Programm also, dessen Stärken sich im Spielgeschehen äußern und nicht im Drumherum.

(hs)

AMIGA DOS
Blitzlicht

Name: Hot Rod
Hersteller: Activision
Quelle: Fachhandel
Preis: 84,95 DM

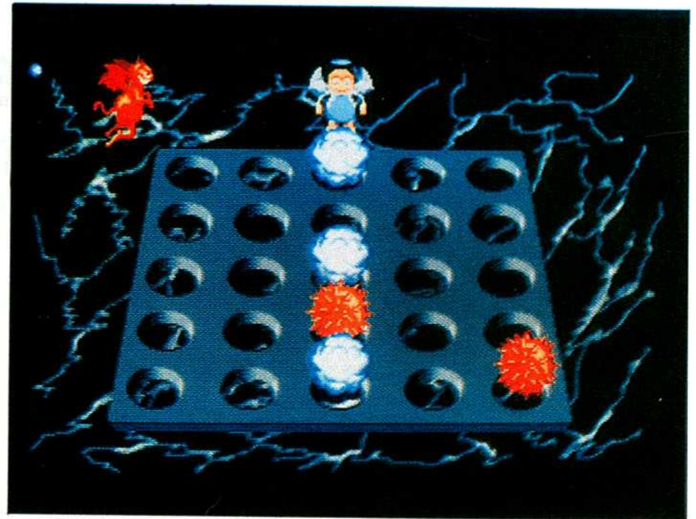
Positiv:
- bis zu vier Spieler
- unterschiedliche Strecken

Negativ:
- anfangs bisweilen unübersichtlich





A.M.C. – ein neues Science-fiction-Spektakel mit vielen fiesen Aliens



Antago – ein Engelchen und ein Teufelchen im Widerstreit

Demnächst auf Ihrem Computer

Buntes Actiongetümmel ist nach wie vor gefragt, aber auch Sport und Simulation erfreuen sich regen Interesses. Aber es zeigt sich auch, daß Totgesagte länger leben. Einer unserer Preview-Titel stellt unter Beweis, daß auch die Textadventures noch lange nicht tot sind.

A.M.C.

Große Spielfiguren, die sich gegen ebenso große Untiere zur Wehr setzen müssen und dabei ein scrollendes Szenario durchwandern: Spiele, die so aussehen, gibt es zuhauf. A.M.C. (was immer das heißen soll...), ein neues Spiel des spanischen Labels Dinamic, ist voll von Rieseninsekten, menschenfressenden Würmern und anderen Monstern, die mit einer großkalibrigen Strahlenwaffe gegrillt werden müssen.

Antago

Ein Engelchen und ein Teufelchen ringen um die Herrschaft auf verschiedenen schwebenden Inseln. Es handelt sich hierbei um eine orginelle Variante des Themas 'Vier gewinnt', nur daß es hier darum geht, fünf Steine in eine Reihe zu gruppieren. Besonders die nette Animation der beiden Hauptfiguren gefällt.

Skidz

Quer durch alle nur denkbaren Teile einer Stadt darf der Spieler mit Skateboard oder Mountain-Bike radeln oder rollern. Zur Schulung des Umweltbewußtseins gilt es, unterwegs Müll einzusammeln und in Papierkörben zu deponieren. Dabei dürfen natürlich keine Passanten oder Tiere über den Haufen gefahren werden, denn dann muß auch der Spieler eine Pause in Kauf nehmen, die das ohnehin knappe Zeitlimit noch weiter verkürzt.

Island of Lost Hope

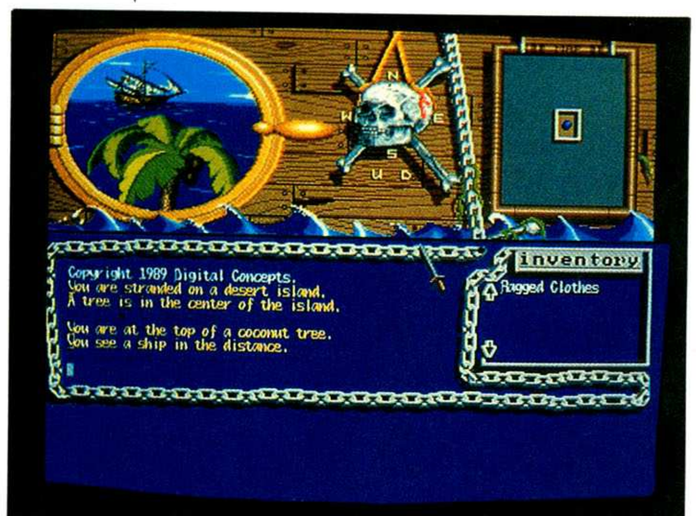
Hier kommen die Freunde der guten alten Textadventures auf ihre Kosten. Aber mit einem einfachen Textspiel wollten die Programmierer sich wohl auch nicht zufrieden geben. So findet der Spieler neben einer Piratenstory auch Animationen und Soundeffekte, die zu den jeweiligen Schauplätzen passen.

Durch Meeresrauschen und Vogelschreie wird der Abenteuerspaß noch intensiver gestaltet. Am Bildschirm wird während des Spiels eine kleine Karte eingeblendet, so daß Verirren eigentlich nicht möglich ist. Der Schwierigkeitsgrad ist moderat, so daß auch Einsteiger eine Chance haben.

(hs)



Skidz – mit Karacho durch die Stadt, mit Skateboard oder Mountain Bike auf großer Fahrt



Island of Lost Hope – die Wiedergeburt des Textadventures

Impressum

Herausgeber
Christian Widuch
Redaktionsleitung
Stefan Ritter
Chefredaktion
Markus Matejka (mm)
Leitender Redakteur
Jürgen Borngießer (jb)
Redaktion
Bernhard Rinke (br), Heinrich Stiller (hs), Joachim Freiburg (jf), Claus Daschner (cd), Vera Brinkmann (vb)
Freie Autoren dieser Ausgabe
Marshal M. Rosenthal, Edgar Meyzis, Bernd Rudolf, Robert Marz, Jürgen Seibel, Garry Glendown, Ulf Petersen, Michael Anton, Andreas Polk, Ingmar Reyer, Timo Siebert, Robert Münch, Fritz Cizmarov, Heiko Schmidt, Michael Herbst, Torsten Budesheim, Karsten Krause, Peter Woop
Redaktionsassistenz
Anke Kerstan (ke), Susanne Eska (es)
Chef vom Dienst
Matthias Bloß
Produktionsleitung
Gerd Köberich
Bereichsleitung
Claudia Ebbrecht (Fotosatz/Lektorat), Margarete Schenk, Helmut Skoupy (Montage/Reprografie)
Layout
Michael Grebenstein
Fotografie
Christian Heckmann
Fotosatz
Gabriela Joseph
Lektorat
Susanne Lessinger
Montage/Reprografie
Manuela Eska, Peter Gajewski
Werbegestaltung
Mohamed Hawa, Petra Küch
Anzeigenleitung
Wolfgang Brill
Anzeigenverkauf
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon (0 89) 4 39 10 87, Telefax 0 89/4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle
Anzeigenpreise
Es gilt die Anzeigenpreisliste Nr. 1 vom 01.05.1990
Anschrift Verlag/Redaktion:
DMV Daten & Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege, Telefon (0 56 51) 8 09-0,
Telefax (0 56 51) 8 09-3 33

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM), Friedrich-Bergius-Straße 20, 6200 Wiesbaden
Druck
Druckerei Jungfer, 3420 Herzberg
Bezugspreise
»AMIGA DOS« erscheint monatlich.
Einzelpreis DM 6,50/sfr. 6,50/öS 52, –
Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland:
12 Ausgaben: DM 70, –
6 Ausgaben: DM 35, –
Europäisches Ausland:
12 Ausgaben: DM 100, –
6 Ausgaben: DM 50, –
Außereuropäisches Ausland:
12 Ausgaben: DM 120, –
6 Ausgaben: DM 60, –
Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 23043-608
Raiffeisenbank Eschwege:
BLZ: 522 603 85, Kto.-Nr.: 245 7008
Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

SPIELEN SIE GERNE?

Lieben Sie es, in finsteren und unheimlichen Gewölben den letzten Geheimnissen auf den Grund zu gehen? Zeichnen Sie gerne Karten von Gewölben und Spiellandschaften?

Dann sind Sie der richtige Partner für uns.

Denn für unsere AMIGA-DOS-Spieletips suchen wir ständig Tips, Tricks, Karten, Cheatmodes und alles andere, was das Spielerherz höher schlagen läßt. Wenn Sie interessante Informationen, Karten oder Lösungsvorschläge für Computerspiele herausgefunden haben, sind Sie unser/e Mann/Frau.
Durch Ihre Information können Sie vielen Spielern weiterhelfen, die ohne Ihre Hilfe unter Umständen monatelang an der gleichen Problematik "festhängen". Oft sind es nur ein Wort oder ein kleiner Hinweis, die dem Spielgeschehen neues Leben einhauchen.

Aus diesen Gründen möchten wir Sie um Ihre Mitarbeit bitten. Wenn Sie also über Spieletips – gleich welcher Art – verfügen, würden wir uns freuen, von Ihnen Post zu bekommen.

Und hier nun die Adresse, an die Sie Ihre Spieletips schicken können:

DMV-Verlag
Redaktion AMIGA DOS
Stichwort: Spieletips
Postfach 250
3440 Eschwege

PS: Unter allen Einsendungen verlosen wir jeden Monat fünf Softwaretitel für die besten Spieletips.

Die Inserenten

Im nächsten Heft

■■■ Musik, Musik, Musik

Das Titelthema unserer nächsten Ausgabe rankt sich rund um Musik und Amiga.

■■■ Modula-Kurs für Ein- und Umsteiger

Wir führen Sie etwas tiefer in die Grundlagen dieser strukturierten Sprache ein.

■■■ MW-500

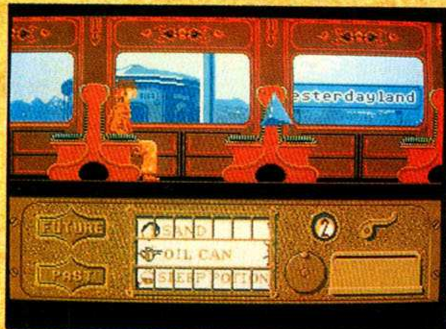
Lang angekündigt, endlich ist es da: das MW-500, der Umbausatz für den A500.

■■■ Lichtmaschine

Wir zeigen Ihnen, wie man den Parallelport des Amiga für eigene elektronische Schaltungen nutzen kann.

■■■ Hard-Work

nennt sich eine Sokoban-Umsetzung, die wir Ihnen im Listingteil vorstellen möchten.



Theme Park Mystery – abenteuerliche Erkundungen auf einem unheimlichen Rummelplatz

■■■ Viele Utilities

Wir haben für Sie in der Trickkiste gekramt und halten wieder jede Menge neuer Tips & Tricks zu Amiga-BASIC, CLI, Assembler, Modula2, C und GFA-BASIC für Sie parat.

■■■ Topaktuelle Spiele

Werden Sie Pilot bei den Blue Angels, und erleben Sie die faszinierenden akrobatischen Meisterleistungen in einer Kunstflugstaffel. Begeben Sie sich mit Terry in Cloud Kingdoms auf die Suche nach den magischen Kristallen. Lassen Sie sich in das Netz des Action-Adventures Theme Park Mystery verstricken. Dazu viele Hinweise zu den Spielen in unserer Helpline.

■■■ Eine geballte Ladung...

Listings, Tests und Infos warten auf Sie.



Der 500er im neuen Gewand – das MW-500

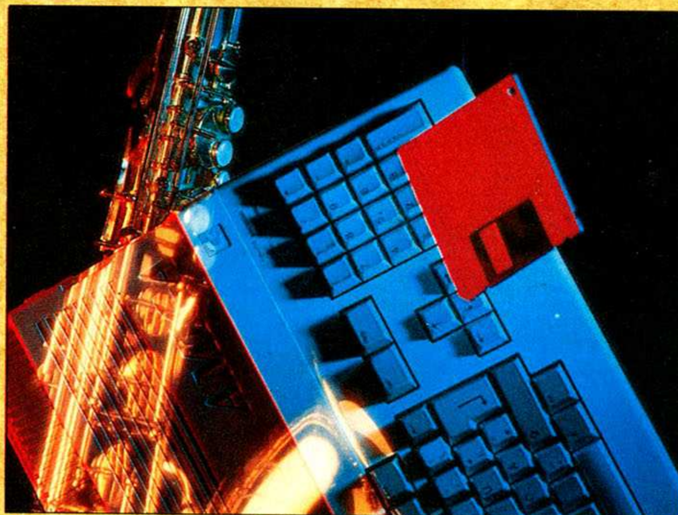
Die nächste

AMIGA DOS

finden Sie ab

11. Juli '90

bei Ihrem Zeitschriftenhändler



Was Sie schon immer über Musik wissen wollten


Amiga
Fraktal Generator 3 D